精通HTML+CSS网页布局与样式

（视频教学版）

一册在手·全面复制

- 全面介绍HTML、CSS、DIV、JavaScript、XML、Ajax网页制作技术
- 各章节教学视频时间长达600分钟
- 下载的代码视频包中包含赠送的3个网站案例及讲解

刘增杰 史艳艳 刘玉萍 编著



清华大学出版社

精通 HTML+CSS 网页布局与样式

视频教学版

刘增杰 史艳艳 刘玉萍 编著

清华大学出版社
北 京

内 容 简 介

本书全面介绍 HTML+CSS 网页设计的主要知识和技能，并提供大量的页面实例供读者实践，使读者能在最短的时间内有效掌握 HTML+CSS 网页制作技术。

本书分为 24 章，内容包括 HTML 网页基础、网页文档结构、网页文本与段落、网页色彩和图片、网页超链接、网页表单、网页表格、使用 HTML 创建框架、网页多媒体、CSS 基础语法、用 CSS 丰富文字和段落效果、用 CSS 美化图片效果、CSS 美化背景与边框、CSS 美化超链接和鼠标、CSS 美化表格与表单样式、CSS 制作菜单、CSS 滤镜、CSS 定位与 DIV 布局、CSS 与 JavaScript 的综合应用、CSS 与 XML 的综合应用和 CSS 与 Ajax 的综合应用等。最后通过两个流行网站的设计，使读者进一步巩固所学的知识，提高综合实战能力。

本书适合 HTML+CSS 网页设计初学者，也适合想全面了解 HTML+CSS 技术的网页设计人员阅读。

图书在版编目（CIP）数据

精通 HTML+CSS 网页布局与样式:视频教学版/刘增杰，史艳艳，刘玉萍编著.--北京:清华大学出版社,2013
ISBN 978-7-302-34010-2

Ⅰ.①精… Ⅱ.①刘… ②史… ③刘… Ⅲ.①超文本标记语言—程序设计②网页制作工具 Ⅳ.①TP312 ②TP393.092

中国版本图书馆 CIP 数据核字（2013）第 234322 号

责任编辑：夏非彼
封面设计：王　翔
责任校对：闫秀华
责任印制：

出版发行：清华大学出版社
网　　址：http://www.tup.com.cn，http://www.wqbook.com
地　　址：北京清华大学学研大厦 A 座　　**邮　　编：**100084
社 总 机：010-62770175　　**邮　　购：**010-62786544
投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn
质量反馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
印 装 者：
经　　销：全国新华书店
开　　本：190mm×260mm　　**印　张：**29.5　　**字　数：**708 千字
版　　次：2013 年 11 月第 1 版　　**印　次：**2013 年 11 月第 1 次印刷
印　　数：1~3000
定　　价：65.00 元

产品编号：053929-01

前　言

随着 Web 网站用户页面体验要求的增高，页面前端技术日趋重要，HTML 和 CSS 技术的成熟，使其在前端技术中突显优势，在目前移动计算平台分化发展的形势下，HTML 和 CSS 技术作为各平台统一开发语言，以其标准布局和精美样式，成为 Web 开发众多技术中不可替代的弄潮儿，受到广大技术人员的重视。本书针对 HTML 和 CSS 页面开发技术，全面讲解 HTML+CSS 构建网页的知识和技巧，使读者快速掌握网页布局和样式的美化方法。

本书内容包括 HTML 和 CSS 的基本语法和用法，设置文字、段落、图片、背景、表格、表单和菜单等网页元素的方法，重点介绍如何使用 HTML+CSS 进行页面布局，注重实战操作，使读者在学习页面编程技术的同时，掌握 HTML 和 CSS 的精髓。除此以外，还介绍了一些扩展知识，包括 CSS 和 JavaScript、XML、Ajax 等综合应用，使读者能够快速入门。最后给出两个综合实例，使读者进一步巩固所学的知识，提高综合实战能力。

本书特色

内容全面：涵盖了所有 HTML+CSS 网页设计知识点，由浅入深地讲解 HTML+CSS 网页设计技术。

图文并茂：注重操作，图文并茂，在介绍案例的过程中，每一个操作均有对应的插图。这种图文结合的方式使读者在学习过程中能够直观、清晰地看到操作的过程以及效果，便于更快地理解和掌握。

案例丰富：把知识点融汇于案例实训当中，并且结合经典案例进行讲解和拓展，使读者达到"知其然，并知其所以然"的效果。

提示技巧：对读者在学习过程中可能会遇到的疑难问题以"提示"和"技巧"的形式进行了说明，以免读者在学习的过程中走弯路。

技术实用：所有案例都是模仿现实网站开发而设计的，在实际工作中的参考性比较强。

读者对象

本书是一本完整介绍 HTML+CSS 网页设计的教程，内容丰富，条理清晰，实用性强。

- 对于 HTML+CSS 网页设计初学者，可以快速上手设计精美的网页。
- 对于 HTML+CSS 设计人员，可以快速掌握网页设计技巧。
- 对于 Web 开发人员，可以作为网页设计参考。

代码下载与技术支持

本书代码可以从下面网址下载：

http://pan.baidu.com/s/121CCt

本书教学视频可以下面网址下载：

http://pan.baidu.com/s/1cdceO

如果代码下载有问题，请发邮件给 booksaga@163.com，邮件主题为“求代码，精通 HTML+CSS”。另外，读者在阅读本书过程中如有疑问可以联系 QQ：443692455。

致谢

本书作者长期从事网站开发实训的培训工作。参与本书编写人员除了封面署名人员以外，还有张俊斌、王英英、吴志才、姬远鹏、张少军、苏士辉、肖品、孙若淞、宋冰冰、王攀登、张工厂、王婷婷、王维维、梁云亮、陈伟光、臧顺娟、程铖、卢健良、李亚飞、李清海、王鹏程、王雪涛、邴万强、陈彭涛、冯玲、郭丽娟、钱东省、王飞、原杨、李坤、武炜、史艳艳、包惠利等人。虽然倾注了编者的努力，但由于水平有限，书中难免有疏漏之处，请读者谅解，如果遇到问题或有意见和建议，敬请与我们联系，我们将全力提供帮助。

编　者

2013 年 8 月

目　　录

第 1 章
网页设计基础知识

在网络高速发展的今天，网站已经成为一个展示自我、宣传自我的工具。公司或个人可以拥有自己的网站，向别人介绍自己，展示自己的风采。这些都离不开 HTML 网页，网页是这些信息的载体。

1.1　认识网页

当打开一个网站后，首先呈现在读者面前的就是网页，网页上的图片、文字、音频和视频等构成网站的基本元素，网页是承载各种网站应用的平台。

1.1.1　网页和网站

网页（Web Page）是一个文件，它存放在世界上某个角落的某一部计算机中，而这部计算机必须是与互联网相连的。网页是由网址（URL，如 www.sohu.com）来识别与存取的，在浏览器中输入网址后，经过一段复杂的程序，网页文件会被传送到读者的计算机中，然后再通过浏览器解释网页内容，将其展示到读者的眼前，网页通常是 HTML 格式的（文件扩展名为.html 或.htm）。

网站（Website）是指在因特网上，根据一定的规则，使用 HTML 等工具制作的用于展示特定内容的相关网页的集合，常见的网站有搜狐、新浪等。简单地说，网站是一种通信工具，就像布告栏一样，人们可以通过网站来发布自己想要公开的资讯信息，或者利用网站来提供相关的网络服务。衡量一个网站的性能时通常从网站空间大小、网站位置、网站连接速度（俗称"网速"）、网站软件配置、网站提供的服务等几方面考虑，最直接的衡量标准是这个网站的真实流量。

在一个网站中，网页按照类型，可以分为两种，即主页和普通网页。主页（Home Page)它是进入一个网站的开始画面，如图 1-1 所示。



图 1-1　搜狐的主页

1.1.2　网页基本构成元素

在 Internet 早期，网站只能保存单纯文本。经过近几年的发展，图像、声音、动画、视频和 3D 等技术在因特网上广泛应用，网站已经发展成图文并茂的样子，并且通过动态网页技术，用户可以与其他用户或者网站管理者进行交流。

网页常见的构成元素有文本、图像、超链接、表格、表单、导航栏、动画、框架等。如图 1-2 所示。



图 1-2　网页常见构成元素

- 文本。网页中的信息主要以文本为主。在网页中可以通过字体、大小、颜色、底纹、边框等来设计文本的属性。
- 图像。有了图像，才能看到丰富多彩的网页。网页上的图片为 JPG 或 GIF 格式。通常图片会被运用在 Logo、Banner 和背景图上。Logo 是代表企业形象或栏目内容的标志性图片，一般在网页的左上角。Banner 是用于宣传网站内某个栏目或活动的广告，一般要求制作成动画形式，达到宣传的效果。Banner 一般位于网页的顶部和底部。在网页页面中比较常用的图片还包括背景图，但要慎用背景图，除非设计者自信背景图可以给网页增加不少魅力。
- 超链接。超链接是网站的灵魂，从一个网页指向另一个目的端的链接。目的端通常是一个网页，也可以是一幅图片、一个电子邮件地址、一个文件、一个程序或者是本网页中的其他位置。超链接本身可以是文字或者图片。

- 表格。表格是网页中展现数据的主要方式，能够以表的形式显示数据信息。表格也可以用作网页排版，在很长一段时间内，使用表格排版是网站的首选方式。
- 表单。表单是用来收集站点访问者信息的集合。站点访问者填写表单的方式是输入文本、单击单选按钮与复选框，以及从下拉菜单中选择选项。在填好表单之后，站点访问者便送出所输入的数据，该数据会根据网站设计者所设置的表单处理程序，以各种不同的方式进行处理。
- 导航栏。导航栏是一组超链接，一般用于网站各部分内容间相互链接的指引。导航栏可以是按钮或者文本超链接。
- 动画。动画是网页上最活跃的元素，包括 GIF 动画和 FLASH 动画。

网页中除了这些最基本元素外，还包括横幅广告、字幕、悬停按钮、计数器、音频、视频等。

1.2 如何设计网页

作为传播信息的一种载体，网页同其他出版物如报纸、杂志等在设计上有许多共同之处，也要遵循一些设计的基本原则。但是，由于表现形式、运行方式和社会功能的不同，网页设计又有其自身的特殊规律。

1.2.1 网页设计基本原则

网页设计是技术与艺术的结合，内容与形式的统一。它要求设计者必须遵循以下三个主要原则。

1. 主题鲜明

一个网页需要有一个明确的主题，并能够以鲜明的样式吸引其他浏览者的注意。这就要求视觉设计一定要有目标和要求，有明确的主题，并在最短时间内被浏览者接收。

网站主题决定了网页设计风格，只有主题表达清楚了，才能够添加艺术信息。一个网站再美，没有主题信息，也不会有浏览者访问。例如，某个旅游网站所展现的都是旅游方面信息，而不是其他信息，如图 1-3 所示。



图 1-3 主体明确

2．形式与内容统一

有句俗话，叫表里如一，用在网页设计中是非常恰当的。任何一个网站都存在内容和形式，内容就是要表达的对象，形式就是内容的表现方式，例如一则新闻，新闻本身是就是内容，具有极高的价值，而形式就是新闻显示的方式和样式，如文本显示，此时需要配置布局、搭配颜色。

在网页设计中，内容可以说是主题、题材等元素的总和，形式就是它的结构、风格和设计语言。内容主宰形式，形式影响内容。形式必须符合内容，二者才能相得益彰。如果只追求花哨的表现形式以及惊人的冲击效果，或者只是追求内容而缺乏任何的艺术表现，此时网页设计会变得空洞而无力。文字和图形都要表述一个中心，如图 1-4 所示。



图 1-4　内容形式统一

3．强调整体

网页整体效果，即整体风格，是网页设计者必须要追求的。正如一个人不能上身穿西装，下身穿裤头，脚上穿雨靴。网页是用来传播信息的一种媒体。它所承载的信息必须具备一定价值，有一定的内容、主题和意念，才能够被人们所接受。

然而，过分追求整体效果，会牺牲页面个体的灵活性。所以，在强调网页整体设计的同时，还应注意网页个体灵活性，不然网页就会显得呆板、沉闷、枯燥，从而影响访问者的热情或浏览欲望。

1.3.2　网页设计成功要素

各个成功的网站，之所以能够吸引眼球，具有较大的浏览量，必然有一些相同的元素。我们将这些网站的成功元素提取出来，使初学者能够尽快掌握，从而设计出优秀的网页。网站设计成功要素有：整体布局、有价值信息、速度、图形和版面设计、文字可读性、网页标题可读性、网站导航、词语准确。

1．整体布局

网页是访问者最先看到的信息，能够在第一时间引人注意，才能把读者留下来。好的网页应该干净整洁、条理清楚、引人入胜。而不是将读者注意力放到一个杂乱的环境中，过多的闪烁、色彩、下拉菜单框、图片等会将读者注意力分散。

2．有价值信息

无论何种站点，其目的都是给读者提供一定的信息，这样读者才会选择留下来阅读。这些有价值信息，可以是新闻、娱乐信息。如果是企业网站，需要提供关于产品和服务的信息，这些信息必须便于理解和查询。

3．速度

如果不能在 20~30 秒之内打开一个网页，一般人就会没有耐心。因此，在设计网页的过程中，应该确保主页速度尽可能快，最好不要用大的图片。

4．图形和版面设计

图形和版面设计关系到浏览者对网页的第一印象，图形应集中反映主页所期望传达的主要信息。例如下面的网页，其 Logo 放在左上角，并且版面整齐、布局合理，如图 1-5 所示。



图 1-5　版面设计

5．文字可读性

在设计网页中的文字信息时，一定要给文字周围留有足够的空间，这样才能使浏览者感觉不到压抑；另外，文字的颜色最好不要使用紫色、橙色和红色，因为这些颜色会让人眼花缭乱。

另一种能够提高文字可读性的因素是所选择的字体，通用字体最易阅读，特殊字体用于标题效果较好，不适合正文。

6．网页标题可读性

网页标题必须容易阅读，需要为所有标题和副标题设置同一字体，并将标题字体加大一号，所有标题和副标题都采用粗体，这样便于识别标题（如字体加大加粗）和副标题（如粗体，与正文字体大小相同），使浏览者一眼就可以看到要点，以便找出并继续阅读感兴趣的内容。

7．网站导航

由于人们习惯于从左到右、从上到下阅读，所以导航条通常应放置在页面左边，对于较长的页面来说，在最底部设置一个简单导航也很有必要。确定一种满意模式之后，最好将这种模式应用到同一网站的每个页面，这样，浏览者就知道如何寻找信息了。

8．词语准确

一个网站如果只有漂亮的外观而词语错误连篇、语法混乱，同样是失败的，因此，应注意词语的准确无误。

1.2.3 网页设计风格及色彩搭配

网站设计要注意的两个要点，即整体风格和色彩搭配。

1．确定网站整体风格

同一个主题，任何两人都不可能设计出完全一样的网站。风格（style）是抽象的，是指站点的整体形象给浏览者的综合感受。例如，我们感觉网易是平易近人的，迪斯尼是生动活泼的，IBM 是专业严肃的，这些都是网站给人们留下的不同感受。例如下面的网站首页，风格简洁、大方，主题明确，如图 1-6 所示。



图 1-6　整体风格

2．网页色彩搭配

无论是平面设计，还是网页设计，色彩永远是最重要的一环。当距离显示屏较远的时候，人们看到的不是优美的版式或美丽的图片，而是网页的色彩。例如下面的网页，其颜色以浅色为主色调，背景色为青色，从而衬托文字和图片效果，如图 1-7 所示。



图 1-7　色彩搭配

下面介绍一些网页色彩搭配的实战经验。

- 白纸黑字是永远的主题
- 网页最常用流行色

蓝色——蓝天白云，沉静整洁的颜色。
绿色——绿白相间，雅致而有生气。
橙色——活泼热烈，标准商业色调。
暗红——宁重、严肃、高贵，需要配黑和灰来压制刺激的红色。

- 颜色忌讳

忌脏——背景与文字内容对比不强烈，灰暗的背景令人沮丧。
忌纯——艳丽的纯色对人的刺激太强烈，缺乏内涵。
忌花——要有一种主色贯穿其中，主色并不是面积最大的颜色，而是最重要、最能揭示和反映主题的颜色。
忌粉——颜色浅固然显得干净，但如果对比过弱，就会显得苍白无力。
蓝色忌纯，绿色忌黄，红色忌艳。

- 几种固定搭配

蓝白橙——蓝为主调。白底，蓝标题栏，橙色按钮或 ICON 做点缀。
绿白兰——绿为主调。白底，绿标题栏，兰色或橙色按钮或 ICON 做点缀。
橙白红——橙为主调。白底，橙标题栏，暗红或桔红色按钮或 ICON 做点缀。
暗红黑——暗红主调。黑或灰底，暗红标题栏，文字内容背景为浅灰色。

1.3 网页设计师应具备的素质

当打开网站时，首先映入眼帘的是该网页的界面设计，如内容介绍、按钮摆放、文字组合、色彩应用、网页导航等，这些都是网页设计师的工作。作为一个优秀的网页设计师，需要同时具备艺术素质、技能素质和综合素质。

1.3.1 艺术素质

美是包含或体现社会生活的本质、规律，引起人们的特定情感反映的具体形象（包括社会形象、自然形象和艺术形象）。审美能力是个人所具有与审美活动相关的主观条件和心理能力。

作为一名网页设计师，要使自己的网页具有美感，就要平时多积累，在仔细观察的基础上分析美的来源，并灵活地将这种美在自己作品中表现出来。只有这样才能使自己的审美能力达到一定的高度。

网页设计中所蕴含的美，是通过视听来实现的。在网页设计中恰当引导访问者的视（文字、图形、动画）、听（声音）这两种感官，使其获得“美感”，是设计中的关键。例如下面的网站首页，通过色彩、图片和创意将美发挥出来，如图 1-8 所示。

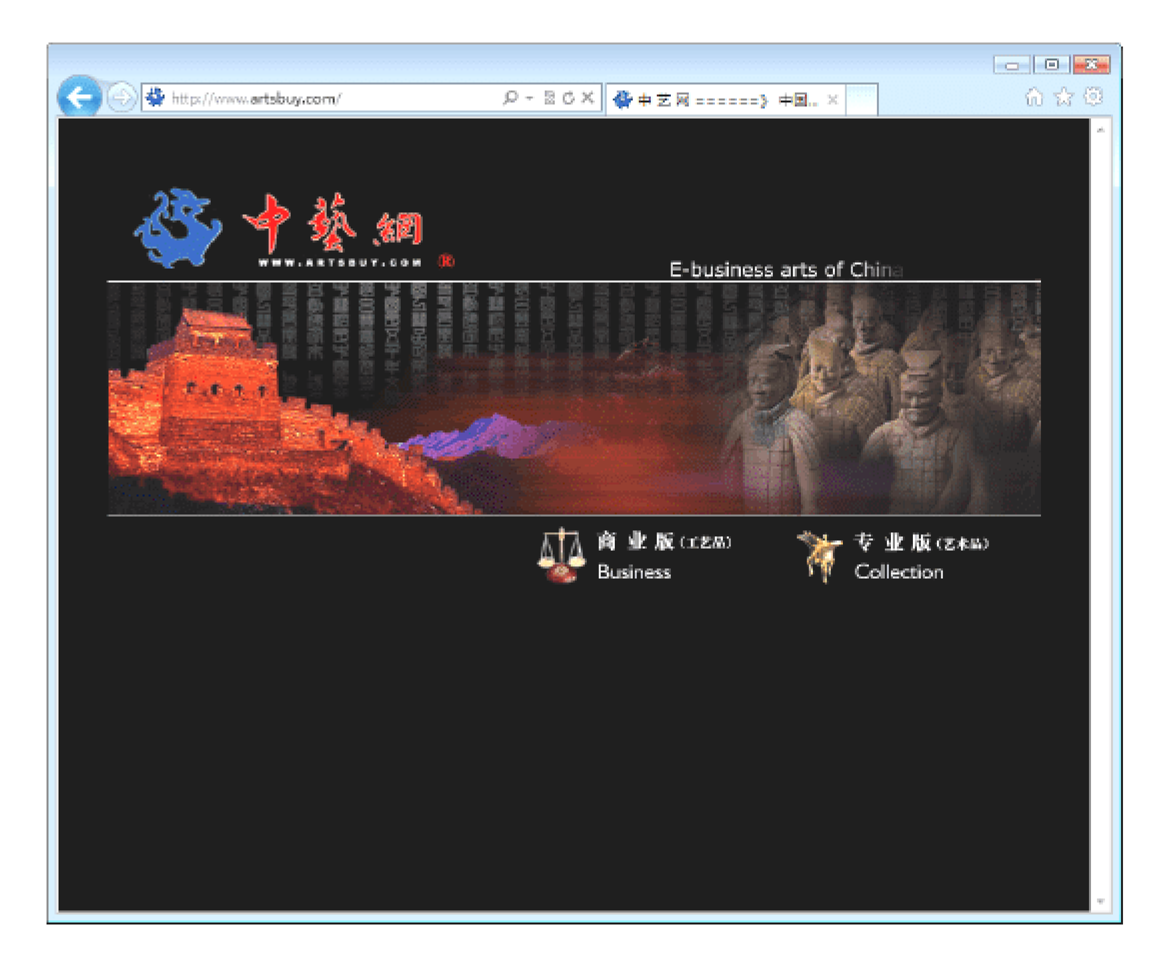


图 1-8　某网站首页

1.3.2　技能素质

优秀的网页设计师应该熟练掌握技术，并在技术和美之间，找到一个结合点，设计出美观而实用的网页。

技术问题是网页设计的核心。网页设计者不仅要考虑设计给用户的感觉、用户访问站点的目的，同时还要考虑自己如何运用技术把思想表达出来，需要哪些工具、哪些素材。这些就要靠自己的日积月累，牢牢地把握住行业的动态。

1.3.3　综合素质

作为合格的网页设计师，一定的文化素质是不可少的。当然文化素质不仅仅包括文学修养，还有音乐和绘画等方面的修养。只有具备这些方面的修养，才能够使自己网页达到一定的水准，才能够制作出令人欣赏的好网页。

网页设计者的素质在一两天内不会有太大的变化，它是随着时间靠辛勤劳动积累下来的。因此，要想成为一名真正的设计高手，网页设计者不仅要有一颗真诚为浏览者服务的心，还要广泛涉猎各个领域的先进成果，特别是要提高自己的艺术修养、文学水平以及鉴赏能力，只要用心了，就离成功不远了。

1.4　专家解惑

1．网页设计中，如何使用图像？

图像内容应有一定的实际作用，切忌虚饰浮夸。图画可以弥补文字的不足，但并不能够完全取代文字。很多用户把浏览软件设定为略去图像，只看文字，以求节省时间。因此，制作主页时，必须注意将图像所连接的重要信息或连接其他页面的指示用文字重复表达几次，同时要注意避免使用过大的图像，如果不得不放置大的图像在网站上，应该把图像的缩小版本的预览效果显示出来，这样用户就不必浪费时间去下载他们根本不想看到的大图像。

2．如何使用动画？

大家都喜欢用 GIF 动画来装饰网页，它的确很吸引人，但在选择时，要确定是否必须使用 GIF 动画，如果回答是否定的，那么就选择静止的图片，因为它的容量要小得多。同样的尺寸的 LOGO，GIF 动画的容量有 5KB，而静止的 LOGO 只有 3KB。虽然只有 2KB 之差，但数量多了，就会影响下载的速度，所以，如果不是必须的，就选择静止图片。

3．网页设计对 HTML 代码要求大么？

为了成功地设计网站，必须理解 HTML 是如何工作的。建议新手应从 HTML 书中去寻找答案。因为用 HTML 设计网站，可以控制设计的整个过程。

第 2 章

HTML网页文档结构

HTML 用标记来表示文档中的文本及图像等元素，并规定浏览器如何显示这些元素，以及如何响应用户的行为。它是标准通用语言（Standard Generalized Markup Language，SGML）的一种应用。本章主要介绍 HTML 的网页文档结构。

2.1　HTML 代码的结构

HTML 语言是用来描述网页的一种语言，它是一种标记语言，而不是编程语言，它是制作网页的基础语言，主要用于描述超文本中内容的显示方式。下面介绍 HTML 代码的基本结构。

2.1.1　HTML 代码的组成

在一个 HTML 文档中，必须包含<html></html>标记，并且放在一个 HTML 文档的开始和结束位置。即每个文档以<html>开始，以</html>结束。<html></hmtl>之间通常包含两个部分，分别为<head></head>和<body></body>，head 标记包含 HTML 头部信息，例如文档标题、样式定义等。body 包含文档主体部分，即网页内容。需要注意的是，HTML 标记不区分大小写。

为了便于读者从整体把握 HTML 文档结构，下面通过一个 HTML 页面来介绍 HTML 页面的整体结构，示例代码如下所示。

```
<html>
<head>
 <title>网页标题</title>
</head>
<body>
  网页内容
</body>
</html>
```

从上面的代码可以看出，基本的 HTML 代码由以下几部分构成。

- <html></html>：说明本页面使用 HTML 语言编写，浏览器软件能够准确无误地进行

解释和显示。

- <head></head>：HTML 的头部标记，头部信息不显示在网页中，此标记内可以存在其他标记。用于说明文件标题和整个文件的一些公用属性。可以通过<style>标记定义 CSS 样式表，通过<script>标记定义 JavaScript 脚本文件。
- <title></title>：title 是 head 中的重要组成部分，它包含的内容显示在浏览器的窗口标题栏中。如果没有 title，浏览器标题栏显示本页的文件名。
- <body></body>：body 包含 HTML 页面的实际内容，显示在浏览器窗口的客户区中。例如，页面中的文字、图像、动画、超链接以及其他 HTML 相关的内容都定义在 body 标记里面。

2.1.2　代码注释

注释是在 HTML 代码中插入的描述性文本，用来解释该代码或提示其他信息。注释只出现在代码中，浏览器对注释代码不进行解释，并且不在浏览器的页面中显示。在 HTML 源代码中适当地插入注释语句是一种非常好的习惯。对于设计者日后的代码修改、维护工作都很有好处。另外，如果将代码交给其他设计者，其他人也能很快读懂前者所编写的内容。

其语法为：

```
<!--注释的内容-->
```

注释语句元素由前后两部分组成，前半部分由一个左尖括号、一个半角感叹号和两个连字符组成，后半部分由两个连字符和一个右尖括号组成。

```
<html>
<head>
<title>标记测试</title>
</head>
<body>
<!-- 这里是标题-->
<h1>HTML 网页文档结构</h1>
</body>
</html>
```

页面注释不但可以对 HTML 中一行或多行代码进行解释说明，而且可以注释掉这些代码。如果希望某些 HTML 代码在浏览器中不显示，可以将这部分内容放在<!--和-->之间，例如，修改上述代码如下。

```
<html>
<head>
<title>标记测试</title>
</head>
<body>
<!--
<h1>HTML 网页文档结构</h1>
```

```
-->
</body>
</html>
```

修改后的代码，将<h1>标记作为注释内容处理，在浏览器中将不会显示这部分内容。

2.2 HTML 的基本标签

标签是 HTML 语言中最基本的单位，每个标签由“<”开始，由“>”结束。标签通过指定某块信息为段落或标题来标识文档某个部件。

2.2.1 HTML 标记<html>

HTML 标记代表文档的开始，由于 HTML 语言语法的松散特性，该标记可以省略，但是为了使之符合 Web 标准和文档的完整性，养成良好的编写习惯，建议不要省略该标记。

HTML 标记以<html>开头，以</html>结尾，文档的所有内容书写在开头和结尾的中间部分。语法格式如下。

```
<html>
...
</html>
```

2.2.2 head 标签及其属性

头标记 head 用于说明文档头部相关信息，一般包括标题信息、元信息、定义 CSS 样式和脚本代码等。HTML 的头部信息以<head>开始，以</head>结束，语法格式如下。

```
<head>
…
</head>
```

<head>元素的作用范围是整篇文档，定义在 HTML 语言头部的内容往往不会在网页上直接显示。

1. 标题标记 title

HTML 页面的标题一般用来说明页面的用途，它显示在浏览器的标题栏中。在 HTML 文档中，标题信息设置在<head>与</head>之间。标题标记以<title>开始，以</title>结束，语法格式如下。

```
<title>
…
</title>
```

在标记中间的“… ”就是标题的内容，它可以帮助用户更好地识别页面。预览网页时，

设置的标题在浏览器的左上方标题栏中显示，此外，在 Windows 任务栏中显示的也是这个标题，如图 2-1 所示。



图 2-1　标题栏在浏览器中的显示效果



页面的标题只有一个，它们位于 HTML 文档的头部，即<head>和</head>之间。

2．元信息标记 meta

<meta>元素可提供有关页面的元信息（meta-information），比如针对搜索引擎和更新频度的描述和关键词。

<meta>标签位于文档的头部，不包含任何内容。<meta>标签的属性定义了与文档相关联的名称/值对，<meta>标签提供的属性及取值如表 2-1 所示。

表 2-1　<meta>标签提供的属性及取值

属性	值	描述
charset	character encoding	定义文档的字符编码
content	some_text	定义与 http-equiv 或 name 属性相关的元信息
http-equiv	content-type expires refresh set-cookie	把 content 属性关联到 HTTP 头部
name	author description keywords generator revised others	把 content 属性关联到一个名称

（1）字符集 charset 属性

在 HTML 中，有一个新的 charset 属性，它使字符集的定义更加容易。例如，下列代码告

诉浏览器，网页使用“ISO-8859-1”字符集显示。

```
<meta charset="ISO-8859-1">
```

（2）搜索引擎的关键字

在早期，Meta Keywords 关键字对搜索引擎的排名算法起到一定的作用，也是很多人进行网页优化的基础。关键字在浏览时是看不到的，其格式如下。

```
<meta name="keywords" content="关键字,keywords" />
```

说明：

- 不同的关键字之间，应用半角逗号隔开（英文输入状态下），不要使用空格或“|”间隔。
- 是“keywords”，不是“keyword”。
- 关键字标签中的内容应该是一个个的短语，而不是一段话。

例如，定义针对搜索引擎的关键字，代码如下。

```
<meta name="keywords" content="HTML, CSS, XML, XHTML, JavaScript" />
```

关键字标签 Keywords，曾经是搜索引擎排名中很重要的因素，但现在已经被很多搜索引擎完全忽略。加上这个标签对网页的综合表现没有坏处，不过，如果使用不恰当的话，对网页非但没有好处，还有欺诈的嫌疑。在使用关键字标签 Keywords 时，要注意以下几点：

- 关键字标签中的内容要与网页核心内容相关，确保使用的关键字出现在网页文本中。
- 使用用户易于通过搜索引擎检索的关键字，过于生僻的词汇不太适合作为 meta 标签中的关键字。
- 不要重复使用关键字，否则可能会被搜索引擎惩罚。
- 一个网页的关键字标签里最多包含 3~5 个最重要的关键字，不要超过 5 个。
- 每个网页的关键字应该不一样。



由于设计者或 SEO 优化者以前对 Meta Keywords 关键字的滥用，导致目前它在搜索引擎排名中的作用很小。

（3）页面描述

Meta Description 元标签（描述元标签）是一种 HTML 元标签，用来简略描述网页的主要内容，通常被搜索引擎用在搜索结果页上显示给最终用户看的一段文字片段。在网页中不显示页面描述，页面描述的使用格式如下。

```
<meta name="description" content="网页的介绍" />
```

例如，定义对页面的描述，代码如下。

```
<meta name="description" content="免费的 web 技术教程。" />
```

（4）页面定时跳转

使用<meta>标记可以使网页在经过一段时间后自动刷新，可通过将 http-equiv 属性值设置为 refresh 来实现。Content 属性值可以设置为更新时间。

我们在浏览网页时经常会看到一些欢迎信息的页面，经过一段时间后，这些页面会自动转到其他页面，这就是网页的跳转。页面定时刷新跳转的语法格式如下。

```
<meta http-equiv="refresh" content="秒;[url=网址]" />
```



上面的[url=网址]部分是可选项，如果有这部分，页面定时刷新并跳转，如果省略该部分，页面只定时刷新，不进行跳转。

例如，要实现每 5 秒刷新一次页面，将下述代码放入 head 标记部分即可。

```
<meta http-equiv="refresh" content="5" />
```

2.2.3　body 标签及其属性

Body 标签定义文档的主体。Body 标签包含文档的所有内容（如文本、超链接、图像、表格和列表等）。网页所要显示的内容都放在网页的主体标记内，它是 HTML 文件的重点。主体标记以<body>标记开始，以</body>标记结束，语法格式如下。

```
<body>
…
</body>
```

注意，在构建 HTML 结构时，标记不允许交错出现，否则会出现错误。

例如，在下列代码中，<body>开始标记出现在<head>标记内。

```
<html>
<head>
<title>标记测试</title>
<body>
</head>
</body>
</html>
```

代码中的第 4 行<body>开始标记和第 5 行的</head>结束标记出现了交叉，这是错误的。HTML 中的所有代码都是不允许交错出现的。

2.3　HTML 文件的编写方法

HTML 文本的编写方法有两种，分别如下：

- 手工编写 HTML 文件。
- 使用 HTML 编辑器。

2.3.1　手工编写 HTML 文件

由于 HTML 是一种标记语言，主要以文本形式存在，因此，所有的记事本工具都可以作为它的开发环境。HTML 文件的扩展名为.html 或.htm，将 HTML 源代码输入到记事本并保存之后，可以在浏览器中打开文档以查看效果。

【例 2.1】使用记事本编写 HTML 文件

具体操作步骤如下。

01 单击 Windows 桌面上的【开始】按钮，选择【所有程序】➢【附件】➢【记事本】命令，打开一个记事本，在记事本中输入 HTML 代码。如图 2-2 所示。

02 编辑完 HTML 文件后，选择【文件】➢【保存】命令或按 Ctrl+S 快捷键，在弹出的【另存为】对话框中，选择【保存类型】为【所有文件】，然后将文件扩展名设为.html 或.htm，如图 2-3 所示。

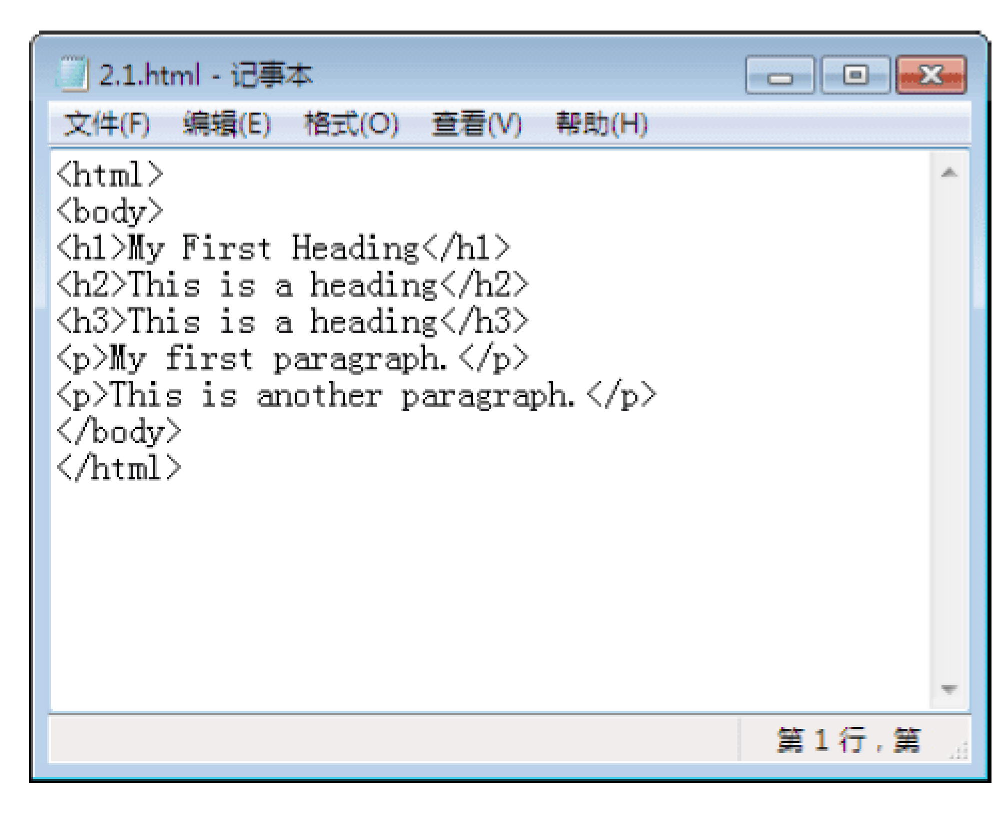


图 2-2　编辑 HTML 代码

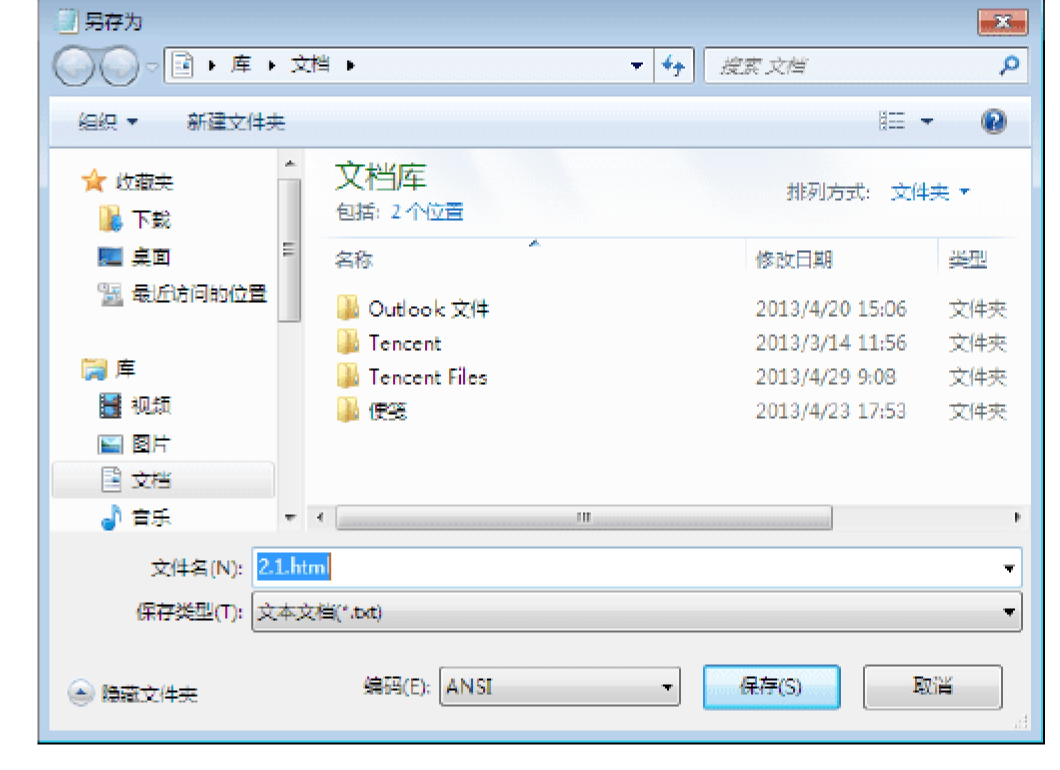


图 2-3　【另存为】对话框

03 单击【保存】按钮，保存文件。打开网页文档，在浏览器中预览效果，如图 2-4 所示。



可以使用记事本编写 HTML 文件，但是编写效率太低，它对于语法错误及格式都没有提示。

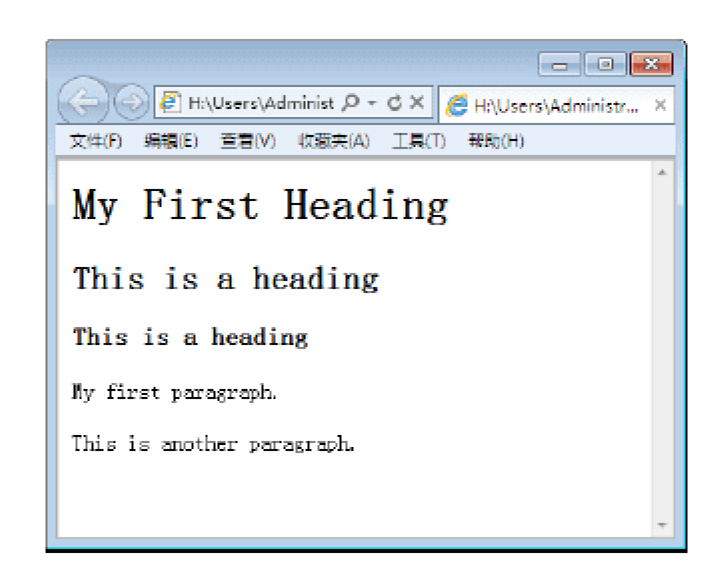


图 2-4　网页的浏览效果

2.3.2　使用 HTML 编辑器

使用 HTML 编辑器可以弥补记事本编写 HTML 文件的缺陷，目前，有很多专门编辑 HTML 网页的编辑器，其中，Adobe 公司出品的 Dreamweaver CS5.5 用户界面非常友好，是一个非常优秀的网页开发工具，并深受广大用户的喜爱。

Dreamweaver CS5.5 中有很多面板，以后在使用时再做详细讲解。打开的面板越多，编辑文档的区域就越小，为了便于编辑文档，可以通过 F4 功能键快速隐藏和显示所有面板。

【例 2.2】使用 Dreamweaver CS5.5 编写 HTML 文件

具体操作步骤如下。

01 启动 Dreamweaver CS5.5，如图 2-5 所示，在欢迎屏幕中【新建】栏中选择【HTML】选项。或者单击菜单中的【文件】➢【新建】命令（快捷键 Ctrl+N）。

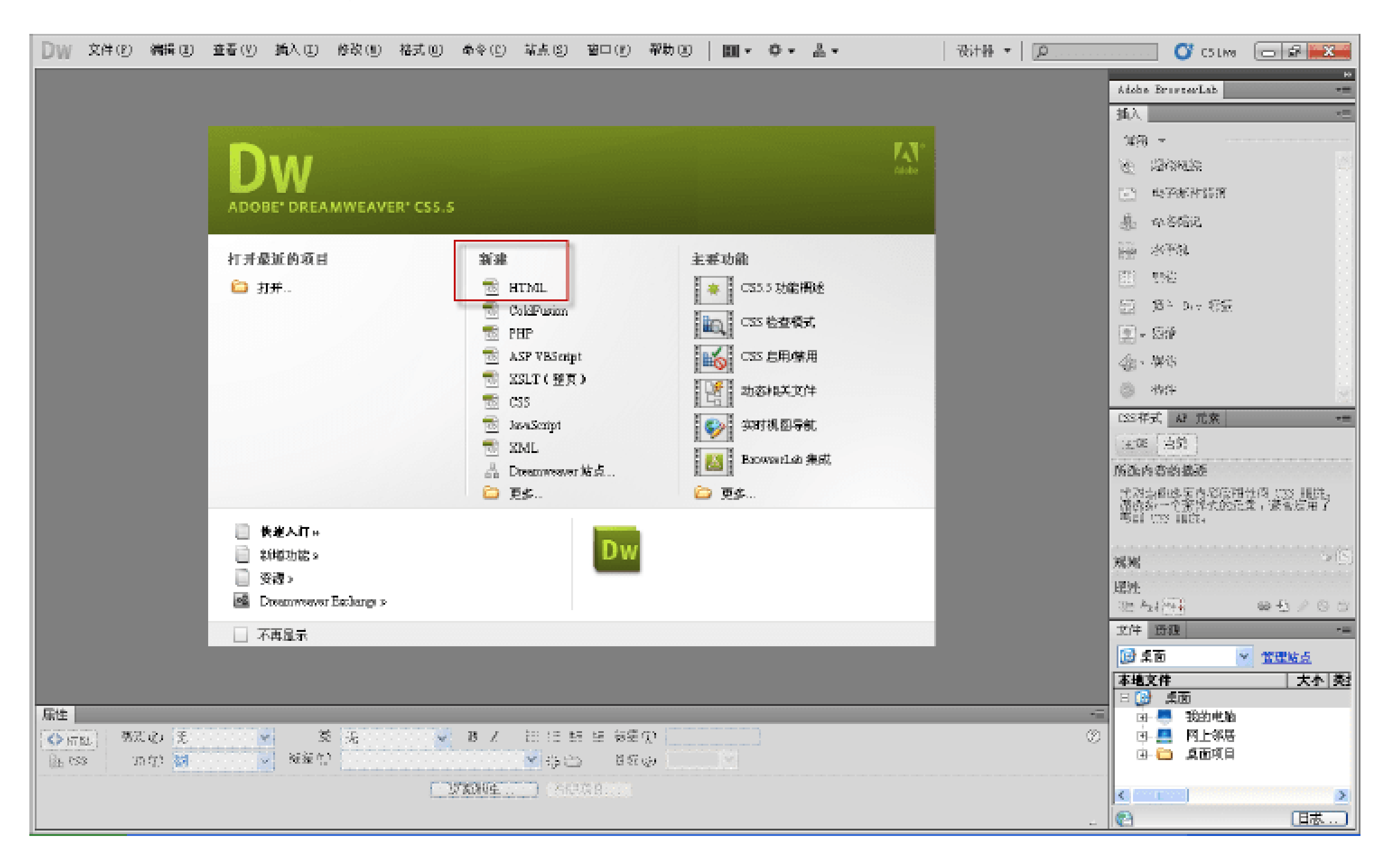


图 2-5　包含欢迎屏幕的主界面

02 弹出【新建文档】对话框，如图 2-6 所示，在【页面类型】选项中，选择【HTML】。

03 单击【创建】按钮，创建 HTML 文件，如图 2-7 所示。

04 在文档工具栏中，单击【代码】按钮，切换到代码视图，如图 2-8 所示。

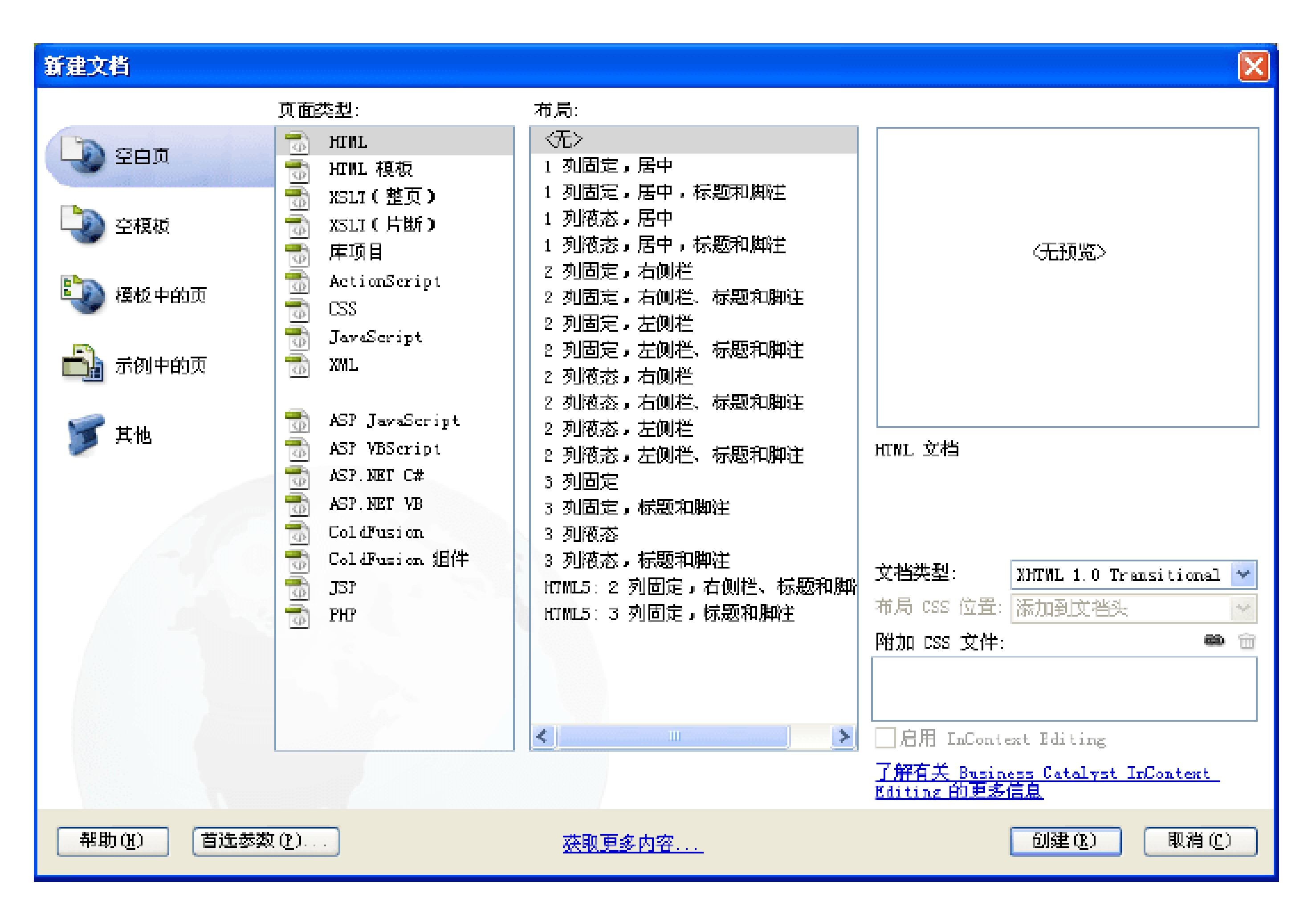


图 2-6　【新建文档】对话框

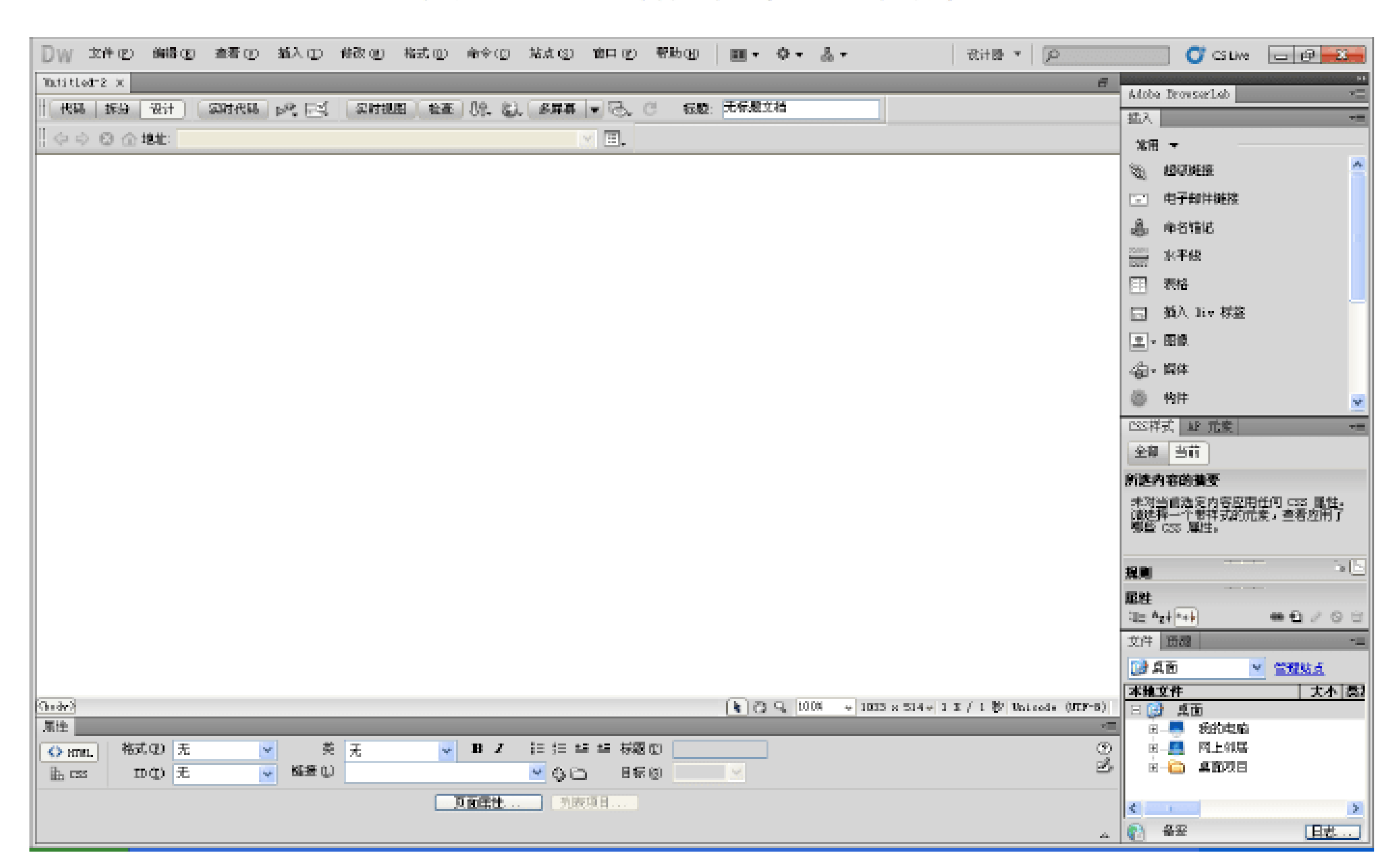


图 2-7　设计视图下显示创建的文档

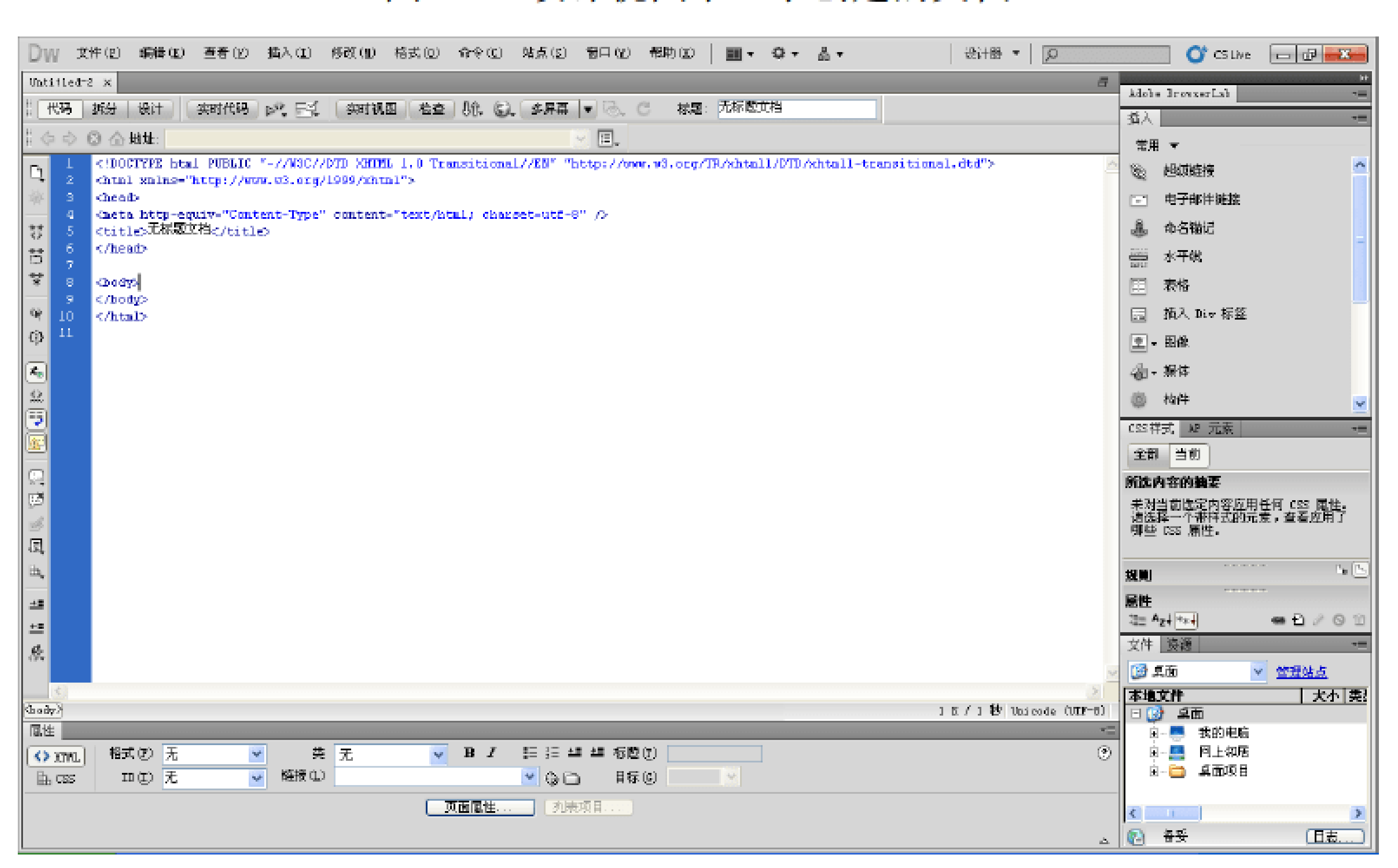


图 2-8　代码视图下显示创建的文档

05 修改 HTML 文档标题，将代码<title>标记中的“无标题文档”修改成“第一个网页”。

06 在<body>标记中键入“今天我使用 Dreamweaver CS5.5 编写了第一个简单网页，感觉非常高兴。”完整的 HTML 代码如下所示。

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>第一个网页</title>
</head>
<body>
今天我使用 Dreamweaver CS5.5 编写了第一个简单网页，感觉非常高兴。
</body>
</html>
```

07 保存文件。单击菜单中的【文件】➢【保存】或按下 Ctrl+S 快捷键，弹出【另存为】对话框。在对话框中，选择保存位置，并输入文件名，单击【保存】按钮，如图 2-9 所示。

08 单击文档工具栏的 图标，选择查看网页的浏览器，或按下 F12 功能键使用默认浏览器查看网页，预览效果如图 2-10 所示。

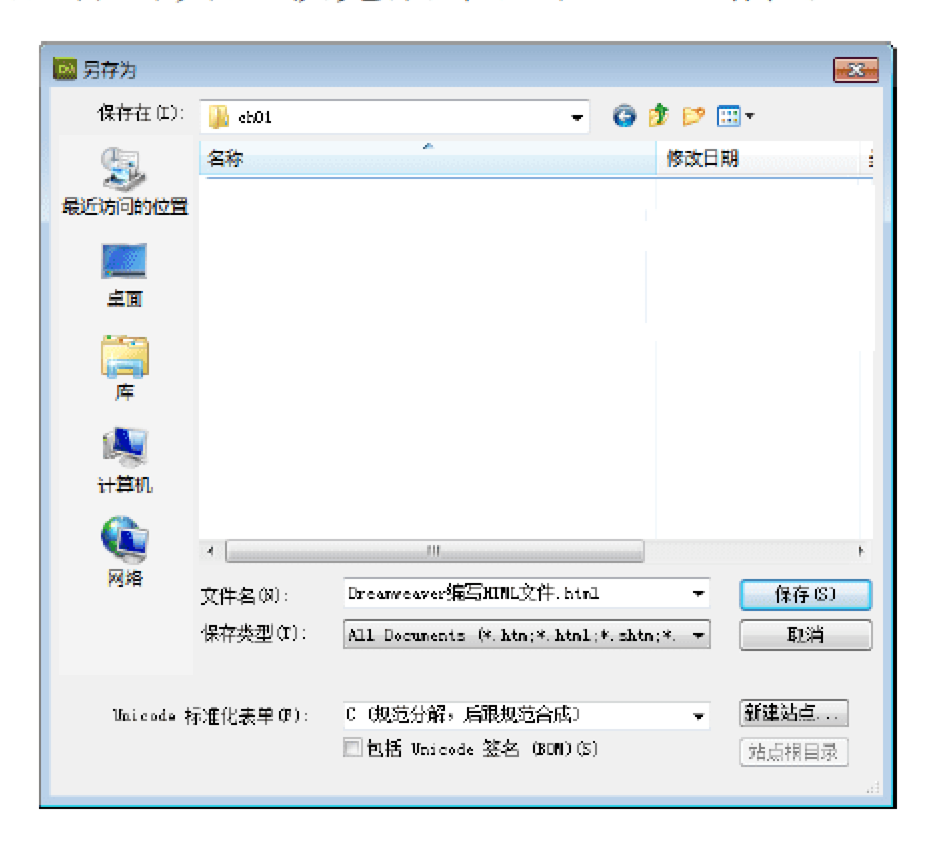


图 2-9　保存文件

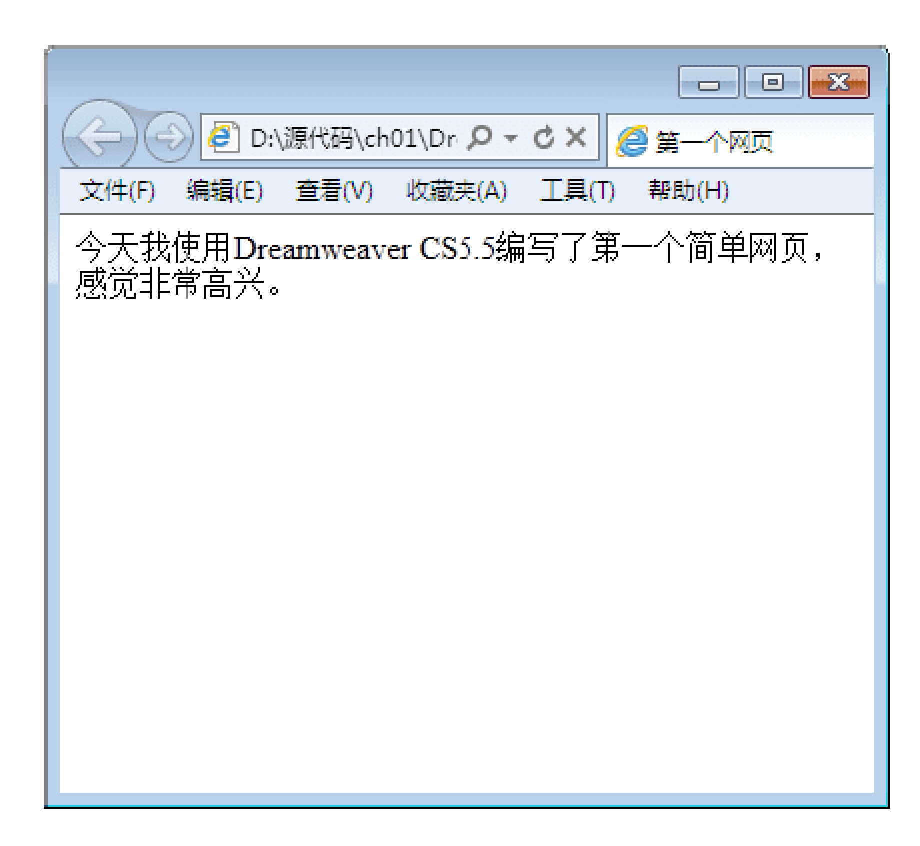


图 2-10　浏览器预览效果

2.4　使用浏览器查看 HTML 文件

浏览器查看效果与 HTML 源代码是开发人员经常使用的。可以使用浏览器查看网页的显示效果，也可以在浏览器中直接查看 HTML 源代码。

2.4.1　各大浏览器与 HTML 的兼容

浏览器是网页的运行环境，因此使用的浏览器类型也是在网页设计时要考虑的一个问题。由于各个软件厂商对 HTML 的标准支持有所不同，导致了同样的网页在不同的浏览器下会有

不同的表现。

另外，对于 HTML 新增的功能，各个浏览器的支持程度也不一致，浏览器的因素变得比以往传统的网页设计更重要。为了保证设计出来的网页在不同的浏览器上的效果一致，本书后面的章节中还会多次提及浏览器。

目前，市面上的浏览器种类繁多，Internet Explorer 是占绝对主流的，因此，本书主要使用 Internet Explorer 9.0 作为浏览器。不过，遇到 IE 浏览器不能支持的情况，将使用 Firefox、Opera 或者其他能支持的浏览器，这点请读者注意。

2.4.2 查看页面效果

双击例 2.2 中编写的 HTML 文件，在 IE 9.0 浏览器窗口中可以看到编辑的 HTML 页面效果，请参阅图 2-10。

前面已经介绍过，网页可以在不同的浏览器中查看，为了测试网页的兼容性，可以在不同的浏览器中打开网页。

在非默认浏览器中打开网页的方法有很多种，在此介绍两种常用方法。

- 方法一：选择浏览器菜单【文件】➢【打开】(有些浏览的菜单项名为“打开文件”)，选择要打开的网页即可。
- 方法二：在 HTML 文件上右击，在弹出的快捷菜单中选择【打开方式】命令，单击需要的浏览器，如图 2-11 所示。如果浏览器没有出现有菜单中，选择【选择程序(C)...】选项，在计算机中查找浏览器程序。



图 2-11　选择不同浏览器打开网页

2.4.3 查看源文件

查看网页源代码的常见方法有以下两种。

- 在打开的页面空白处右击，在弹出的快捷菜单中选择【查看源文件】命令，如图 2-12 所示。
- 在浏览器菜单栏上选择【查看】➢【源文件】菜单命令，可以查看源文件，如图 2-13 所示。

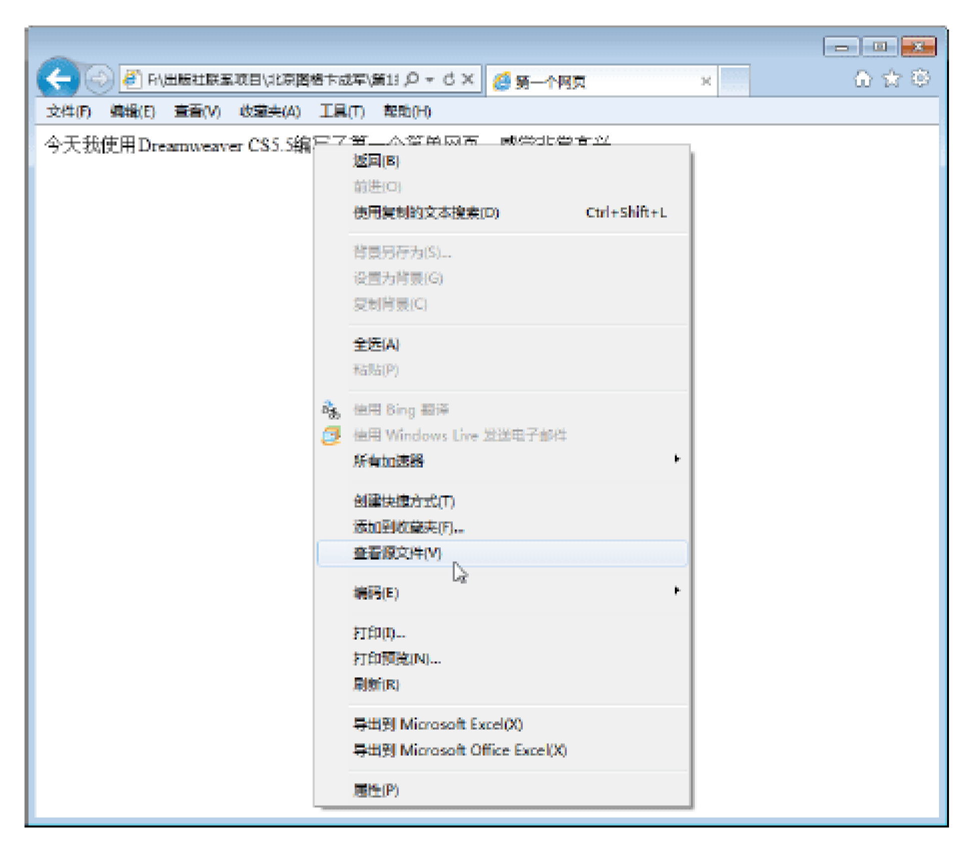




图 2-12 选择【查看源文件】菜单命令

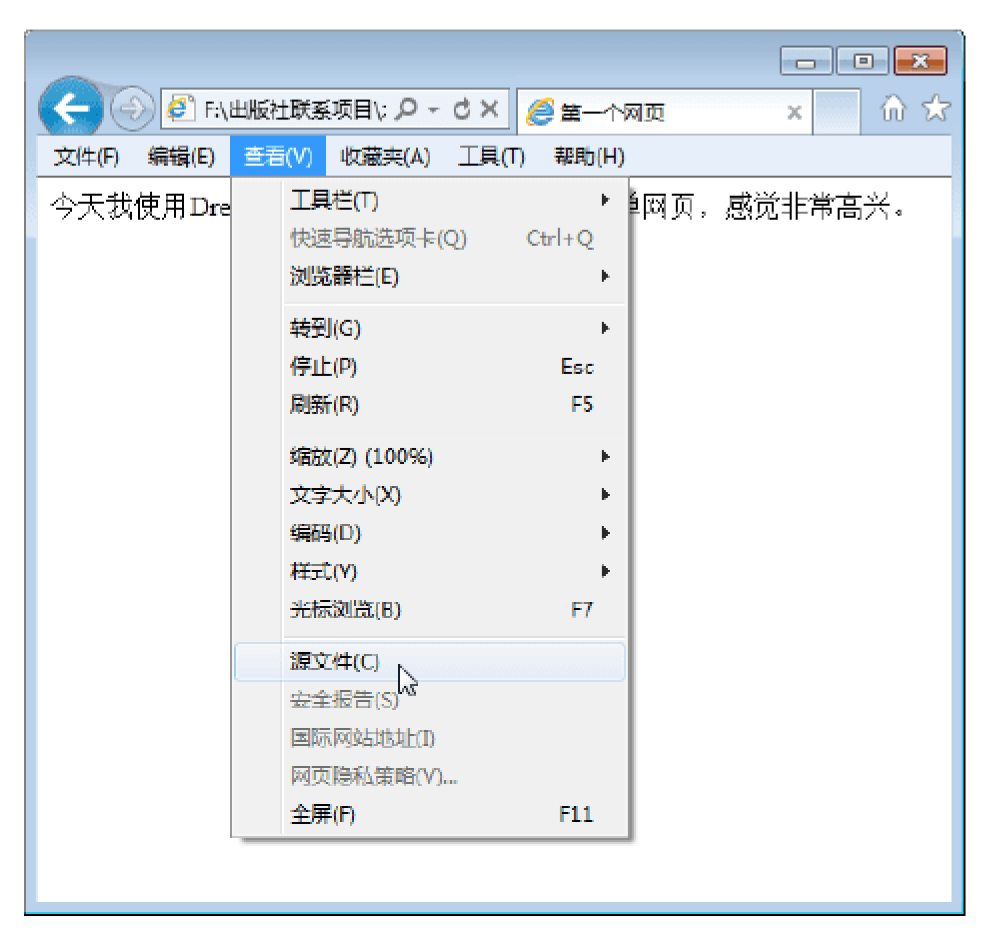


图 2-13 选择【源文件】菜单命令



由于浏览器的规定各不相同，有些浏览器将【查看源文件】命名为【查看源代码】，请读者注意，但是操作方法完全相同。

2.5 专家解惑

1．为何使用记事本编辑的 HTML 文件无法在浏览器中预览，而是直接在记事本中打开？

答：很多初学者在保存文件时，没有将 HTML 文件的扩展名.html 或.htm 作为文件的后缀，导致文件还是以.txt 为扩展名，因此，无法在浏览器中查看。如果读者是通过单击右键来创建记事本文件，在给文件重命名时，一定要以.html 或.htm 作为文件的后缀。特别要注意的是当 Windows 系统的扩展名被隐藏时，更容易出现这样的错误。读者可以在【文件夹选项】对话框中查看是否显示扩展名。

2．如何设置 Dreamweaver CS5.5 欢迎屏幕的显示与隐藏？

答：Dreamweaver CS5.5 欢迎屏幕可以帮助使用者快速进行打开文件、新建文件和相关帮助的操作。如果读者不希望显示该窗口，可以按下 Ctrl+U 快捷键，在弹出的窗口中，选择左侧的【常规】页，将右侧【文档选项】部分的【显示欢迎屏幕】勾选取消。

第 3 章

网页文本与段落的设计

文本是网页中最主要也是最常用的元素。本章主要介绍如何向网页中添加文本，并设计文本段落列表的相关格式，如建立无序列表和有序列表、设置段落格式的显示方式等。

3.1 网页文本的特殊样式

在文档中经常出现重要文本，这些文本将进行特殊显示，如加粗显示、斜体显示、添加上标或下标等。

3.1.1 粗体显示

在网页中，比较重要的文本通常以粗体方式显示。使用 HTML 的<b>标记可以设置文本的粗体显示效果。

【例 3.1】（实例文件：ch03\3.1.html）

```
<html>
<head>
<title>粗体显示</title>
</head>
<body>
<p>清明时节雨纷纷</p>
<p><b>清明时节雨纷纷</b> </p>
</body>
</html>
```

在 IE 9.0 中的预览效果如图 3-1 所示，可以看到第一句为默认显示方式，第二句为粗体显示方式。

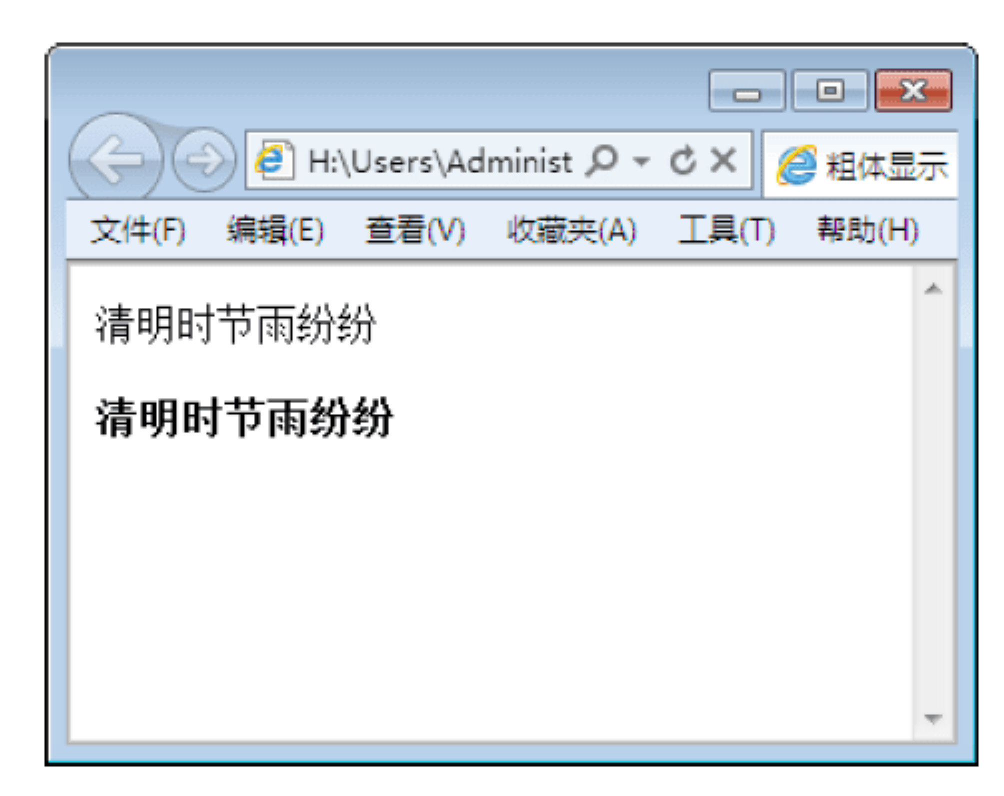


图 3-1　粗体显示预览效果

3.1.2　强调显示

使用<em>标签可以强调显示文本内容，在浏览器中的表现效果就是把这段文字以斜体方式显示。

【例 3.2】（实例文件：ch03\3.2.html）

```
<html>
<head>
<title>粗体显示</title>
</head>
<body>
<p>清明时节雨纷纷</p>
<p><em>清明时节雨纷纷</em> </p>
</body>
</html>
```

在 IE 9.0 中预览效果如图 3-2 所示，可以看到第一句为默认显示方式，第二句为强调显示方式。

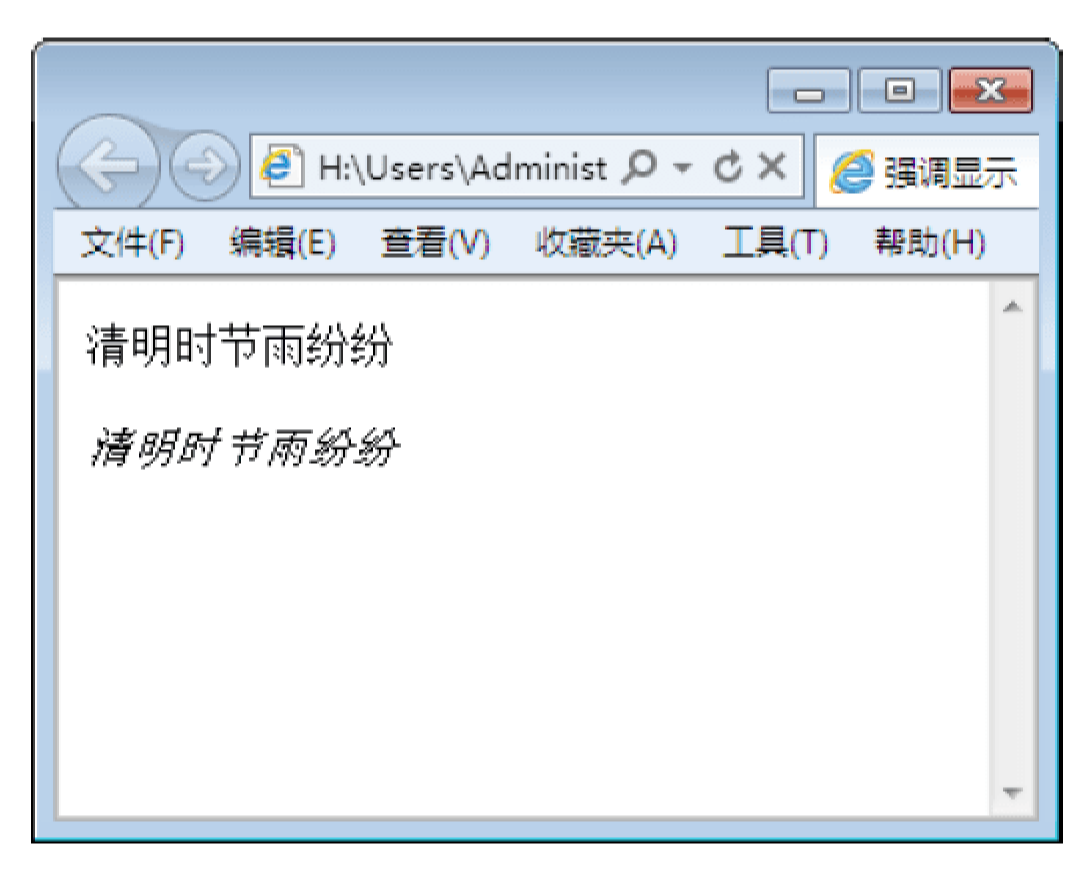


图 3-2　强调显示预览效果



在文本中加入强调也需要有技巧。如果强调太多，有些重要的短语就会被漏掉；如果强调太少，就无法真正突出重要的部分，这与调味品一样，最好还是不要滥用强调。另外，如果只想使用斜体字来显示文本，请使用<i>标签。

3.1.3　加强调显示

<strong>标签和<em>标签一样，用于强调文本，但它强调的程度更强烈一些。浏览器通常会以不同于<em>标签的方式来显示<strong>标签中的内容，通常是用加粗的字体（相对于斜体）来显示其中的内容，这样用户就可以把这两个标签区分开来了。

【例 3.3】（实例文件：ch03\3.3.html）

```
<html>
<head>
<title>无标题文档</title>
</head>
<body>
<p>清明时节雨纷纷</p>
<p><strong>清明时节雨纷纷</strong></p>
</body>
</html>
```

在 IE 9.0 中预览效果如图 3-3 所示，可以看到第一句为默认显示方式，第二句为加强调显示方式。

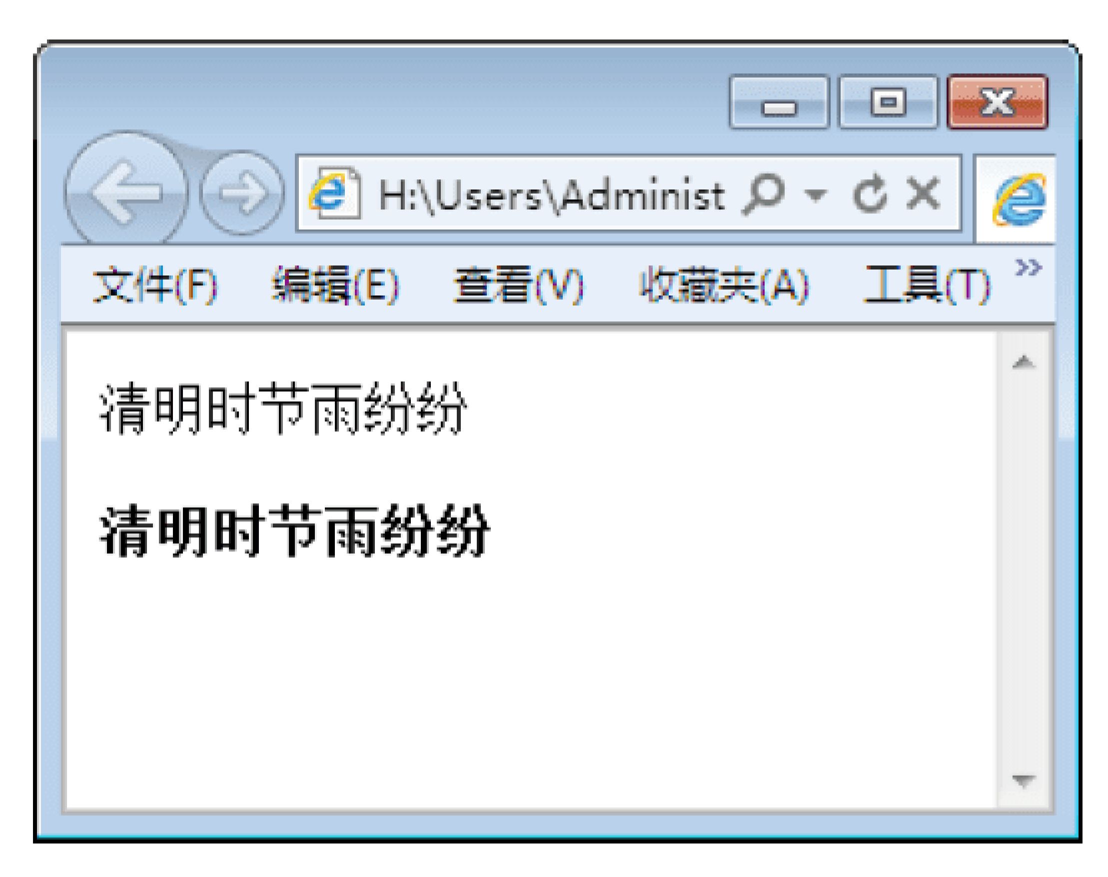


图 3-3　加强调显示预览效果

3.1.4　倾斜文本

HTML 中的<i>标记实现了文本的倾斜显示。放在<i></i>之间的文本将以斜体显示。

【例 3.4】（实例文件：ch03\3.4.html）

```
<html>
```

```
<head>
<title>无标题文档</title>
</head>
<body>
<p>路上行人欲断魂</p>
<p><i>路上行人欲断魂</i></p>
</body>
</html>
```

在 IE 9.0 中预览效果如图 3-4 所示，其中<i><i>之间的文字以斜体显示。

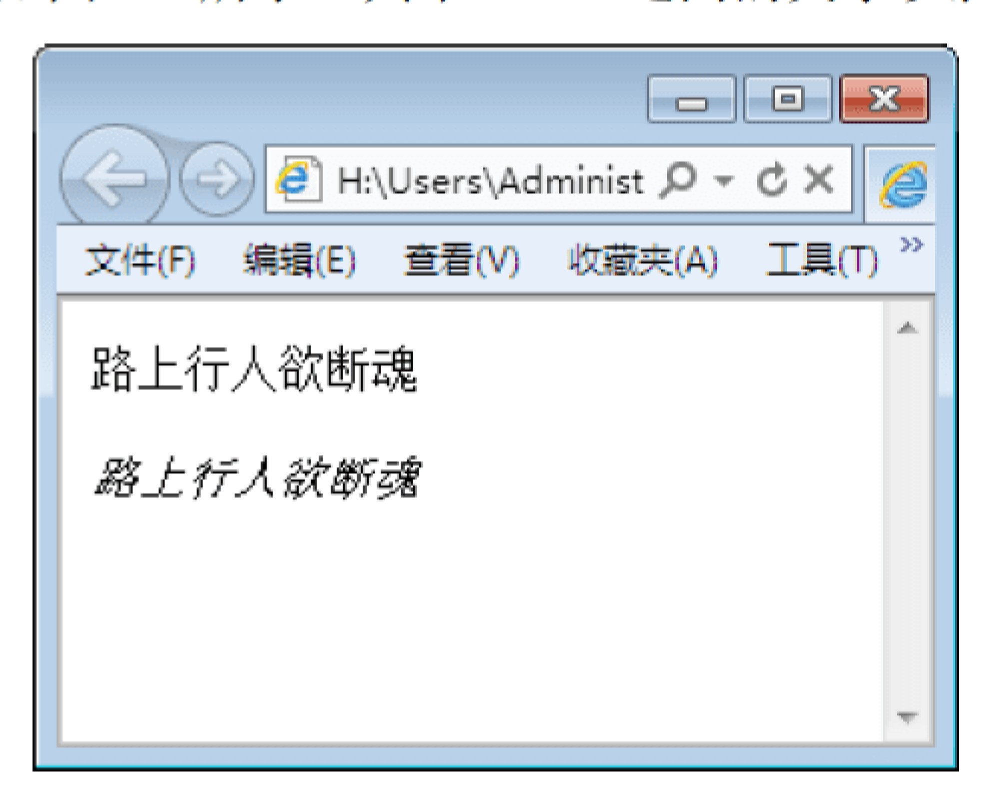


图 3-4　斜体文本预览效果



HTML 中的重要文本和倾斜文本标记已经过时，这些标记都应该使用 CSS 样式来实现，而不应该使用 HTML 来实现。随着后面学习的深入，读者会逐渐发现，即使 HTML 和 CSS 实现了相同的效果，但是 CSS 所能实现的控制远远比 HTML 要细致、精确很多。

3.1.5　上标和下标文本

在 HTML 中用<sup>标记实现上标文字，用<sub>标记实现下标文字。<sup>和<sub>都是双标记，两个标记之间的文本会分别以上标或下标形式出现。

【例 3.5】（实例文件：ch03\3.5.html）

```
<html>
<head>
<title>无标题文档</title>
</head>
<body>
 <!-上标显示-->
 <p>c=a<sup>2</sup>+b<sup>2</sup></p>
<!-下标显示-->
```

```
  <p>H<sub>2</sub>+O→H<sub>2</sub>O</p>
</body>
</html>
```

在 IE 9.0 中预览效果如图 3-5 所示，分别实现了了上标和下标文本显示。

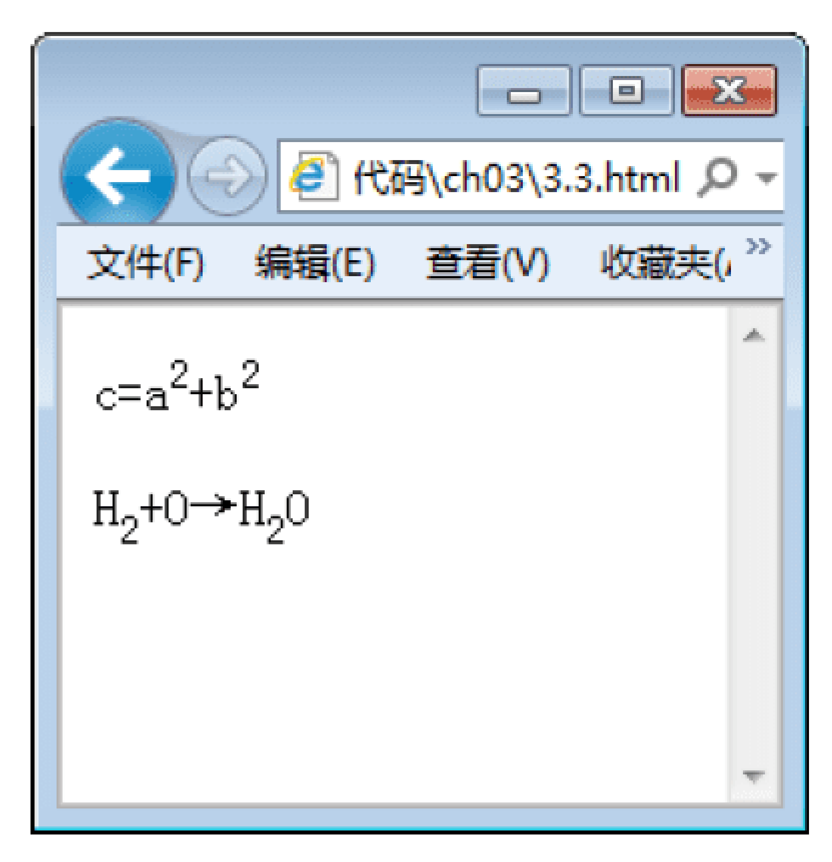


图 3-5　上标和下标预览效果

3.1.6　设置文字大小

在 HTML 中，使用 big 和 small 标签可以定义文字的大小。其中，<big>标签呈现大号字体效果，<small>标签呈现小号字体效果。

【例 3.6】（实例文件：ch03\3.6.html）

```
<html>
<head>
<title>清明</title>
</head>
<body>
<p>清明时节雨纷纷</p>
<p><big>路上行人欲断魂</big></p>
<p><small>借问酒家何处有</small></p>
<p>牧童遥指杏花村</p>
</body>
</html>
```

在 IE 9.0 中预览效果如图 3-6 所示，分别实现字体放大与缩小的显示。



使用<big>标签时要小心，因为浏览器总是试图去理解各种标签，对于那些不支持<big>标签的浏览器来说，它经常将其认为是粗体字标签。另外，如果文字已经是最大号字体，<big>标签将不起任何作用；而如果字体已经是最小号，<small>标签也将不起任何作用。

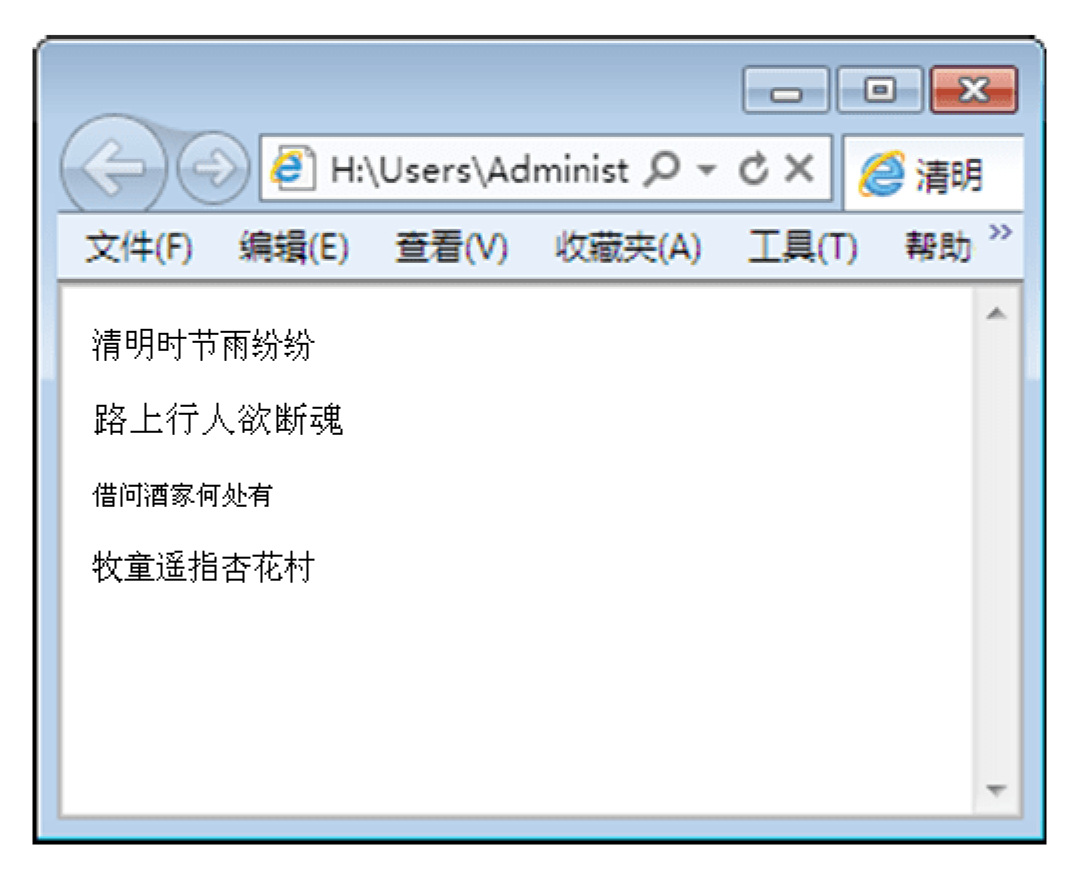


图 3-6　字体放大与缩小显示效果

3.1.7　定义文字的方向

在网页中，为了吸引浏览者的眼球，有时会使文字从右向左显示，这就需要 HTML 的<bdo>标签，不过，<bdo>标签需要与属性 dir 一起使用。其中，属性值“rtl”表示文字从右向左显示，“ltr”表示从左向右显示。

【例 3.7】(实例文件：ch03\3.7.html)

```
<html>
<body>
<p>春眠不觉晓</p>
<bdo dir="rtl">春眠不觉晓</bdo>
<p><bdo dir="ltr">春眠不觉晓</bdo></p>
</body>
</html>
```

在 IE 9.0 中预览效果如图 3-7 所示，在图中可以看到第一句是正常显示，第二句是换方向显示，第三句是使用属性值“ltr”设置的效果，与第一句显示效果一样。

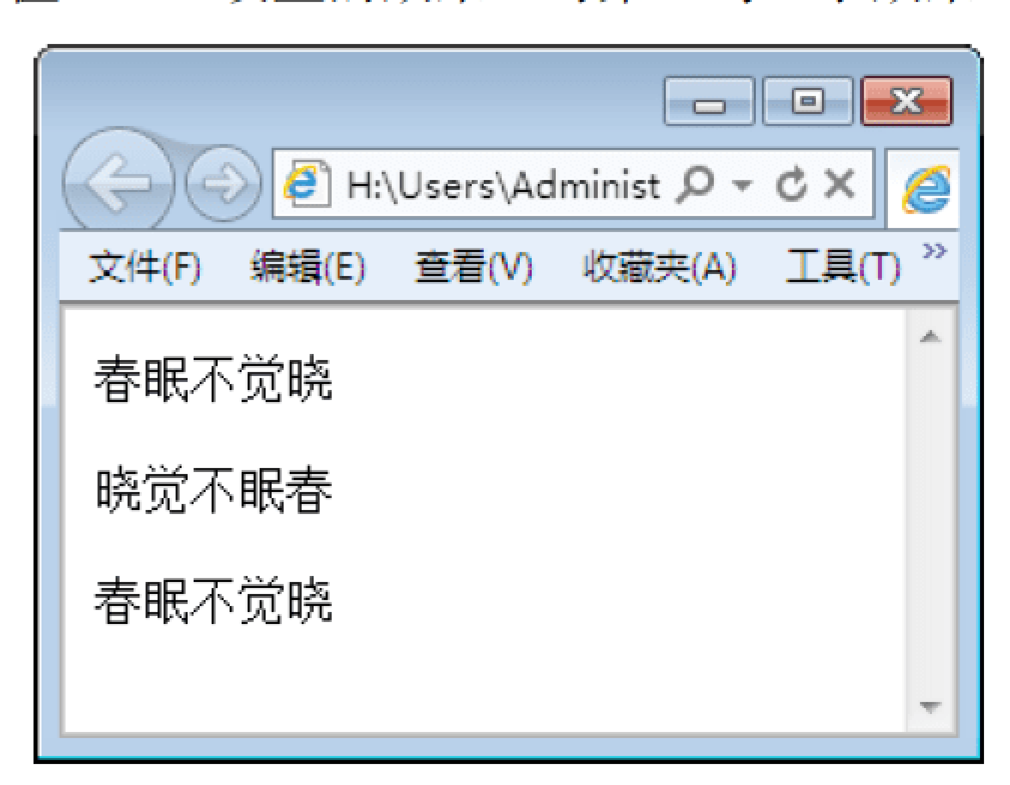


图 3-7　文字方向预览效果

3.1.8 设置文字的字体、大小与颜色

有时，需要对网页中的文字进行字体样式、大小、颜色等属性的设置。在 HTML 中，可以使用<font>标签来设置文字大小、颜色等样式。

【例 3.8】（实例文件：ch03\3.8.html）

```
<html>
<head>
<title>《清明》</title>
</head>
<body>
《清明》
<p>清明时节雨纷纷</p>
<p><font size="5" face="arial" color="red">路上行人欲断魂</font> </p>
<p><font size="6" face="verdana" color="green">借问酒家何处有</font> </p>
<p><font size="7" face="Webdings" color="blue">牧童遥指杏花村</font> </p>
</body>
</html>
```

在 IE 9.0 中预览效果如图 3-8 所示，从图中可以看到设置文字颜色、大小后的效果。

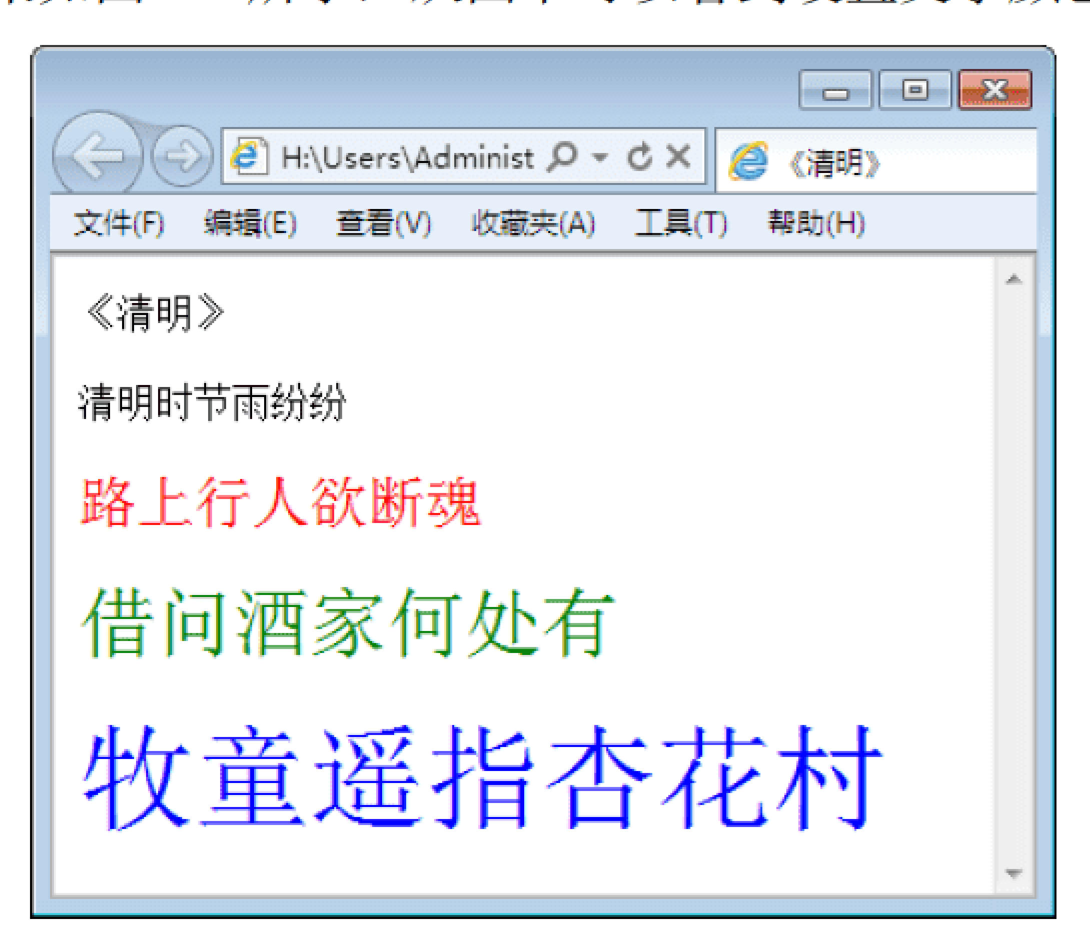


图 3-8 文字样式的预览效果



在 HTML 中，不赞成使用 font 元素。如果需要设置文字的字体、颜色和大小，请使用 CSS 样式。

3.2 网页文本排版

在网页中如果要把文字合理地显示出来，离不开段落标记。对网页中的文字段落进行排版时，并不像文本编辑软件 Word 那样可以定义许多模式来安排文字的位置。在网页中要将某一

段文字放在特定的地方是通过 HTML 标记来完成的。

3.2.1　换行标记

使用换行标记
可以给一段文字换行，该标记是一个单标记，它没有结束标记，是英文单词“break”的缩写，其作用是将文字在一段内强制换行。一个
标记代表一个换行符，连续的多个标记可以实现多次换行。使用换行标记时，在需要换行的位置添加
标记即可。例如，下面的代码实现了对文本的强制换行。

【例 3.9】（实例文件：ch03\3.9.html）

```
<html>
<head>
<title>文本段换行</title>
</head>
<body>
本节目标<br />
网页中的文字是如何设置的<br/>
如何在 Dreamweaver 中处理文字<br/>
如何对文本进行格式化（CSS）<br/>
熟悉使用 Dreamweaver 进行样式表的创建与应用
</body>
</html>
```

虽然在 HTML 源代码中，增加
标记后，在 IE 9.0 中预览效果如图 3-9 所示，实现了换行效果。

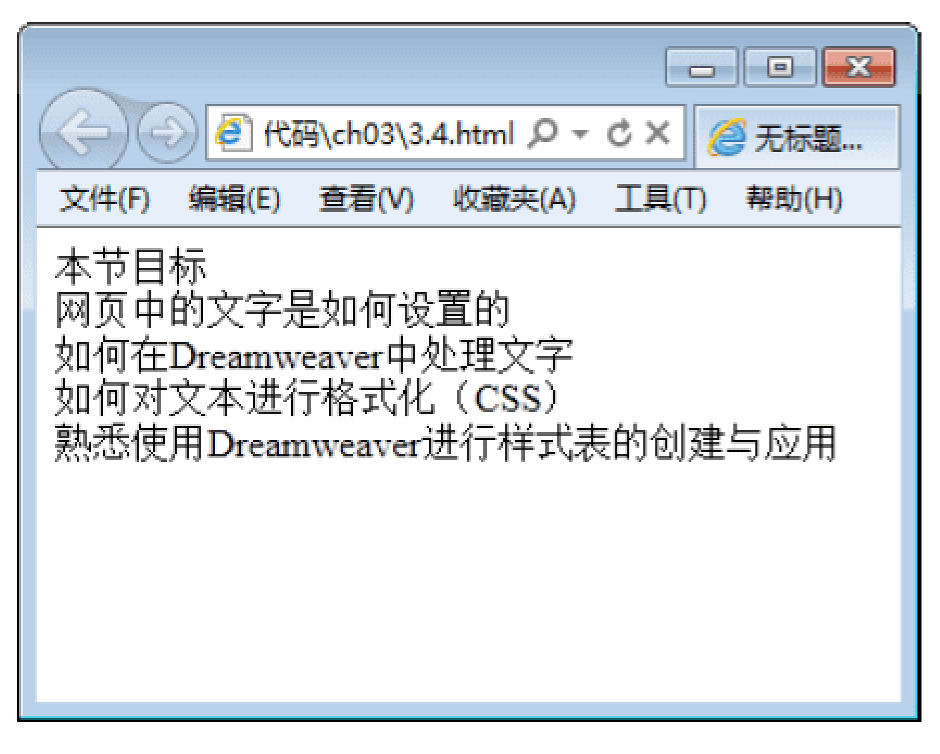


图 3-9　换行标记的使用

3.2.2　段落标记<p>

段落标记<p>用来定义网页中的一段文本，文本在一个段落中会自动换行。段落标记是双标记，即<p></p>，在<p>开始标记和</p>结束标记之间的内容形成一个段落。如果省略结束标记，从<p>标记开始，直到遇见下一个段落标记之前的文本，都在一段段落内。段落标记中的 p 是英文单词“paragraph”的首字母。

【例 3.10】（实例文件：ch03\3.10.html）

```
<html>
<head>
<title>段落标记的使用</title>
</head>
<body>
<p>HTML、CSS 应用教程之 跟 DIV 说 Bey!Bey!</p>
<p>Web 设计师可以使用 HTML4 和 CSS2.1 完成一些很酷的东西。我们可以在不使用陈旧的基于 table 布局的基础上完成文档逻辑结构并创建内容丰富的网站。我们可以在不使用内联<font>和<br>标签的基础上对网站添加漂亮而细腻的风格样式。事实上，我们目前的设计能力已经让我们远离了那个可怕的浏览器战争时代、专有协议和那些充满闪动、滚动和闪烁的丑陋网页。
</p>
<p>
虽然我们现在已经普遍使用了 HTML4 和 CSS2.1，但是我们还可以做得更好！我们可以重组我们代码的结构并能让我们的页面代码更富有语义化特性。我们可以缩减带给页面美丽外观样式代码量并让他们有更高的可扩展性。现在，HTML 和 CSS 正跃跃欲试的等待大家，下面让我们来看看他们是否真的能让我们的设计提升到下一个高度吧…
</p>
<p>
曾经，设计师们经常会跟频繁使用基于 table 的没有任何语义的布局。不过最终还是要感谢像 Jeffrey Zeldman 和 Eric Meyer 这样的思想革新者，聪明的设计师们慢慢的接受了相对更语义化的<div>布局替代了 table 布局，并且开始调用外部样式表。但不幸的是，复杂的网页设计需要大量不同的标签结构代码，我们把它叫做“<div>-soup” 综合症。
</p>
</body>
</html>
```

在 IE 9.0 中预览效果如图 3-10 所示，<p>标记将文本分成 4 个段落。

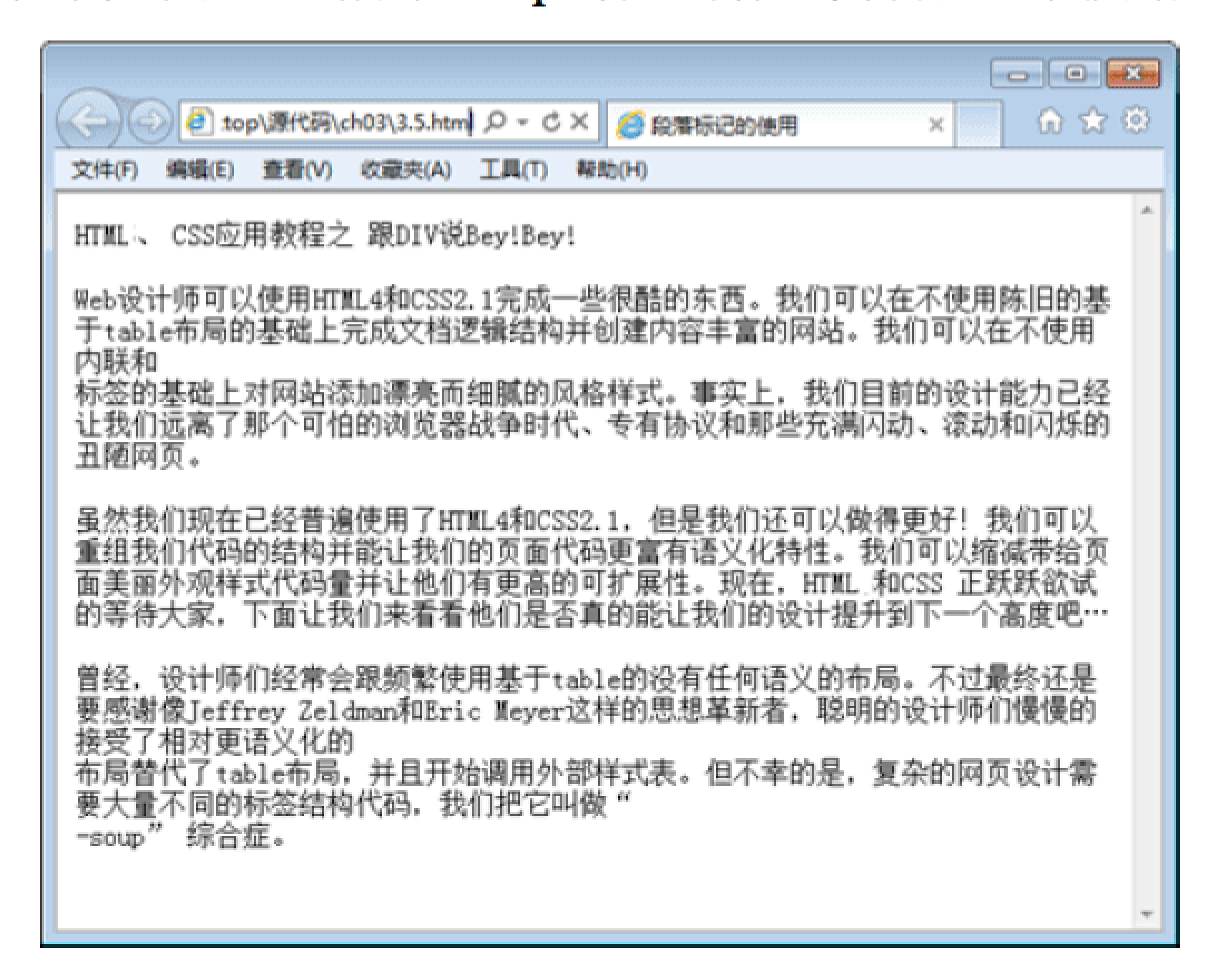


图 3-10　段落标记的使用

3.2.3 网页标题标记<h1>~<h6>

在 HTML 文档中，文本的结构除了以行和段出现之外，还可以作为标题存在。通常一篇文档最基本的结构就是由若干不同级别的标题和正文组成的。

HTML 文档中包含各种级别的标题，各种级别的标题由<h1>~<h6>元素来定义，<h1>~<h6>标题标记中的字母 h 是英文 headline（标题行）的简称。其中<h1>代表 1 级标题，级别最高，字体也最大，其他标题元素依次递减，<h6>级别最低。

【例 3.11】（实例文件：ch03\3.11.html）

```
<html>
<head>
<title>文本段换行</title>
</head>
<body>
<h1>这里是 1 级标题</h1>
<h2>这里是 2 级标题</h2>
<h3>这里是 3 级标题</h3>
<h4>这里是 4 级标题</h4>
<h5>这里是 5 级标题</h5>
<h6>这里是 6 级标题</h6>
</body>
</html>
```

在 IE 9.0 中预览效果如图 3-11 所示。

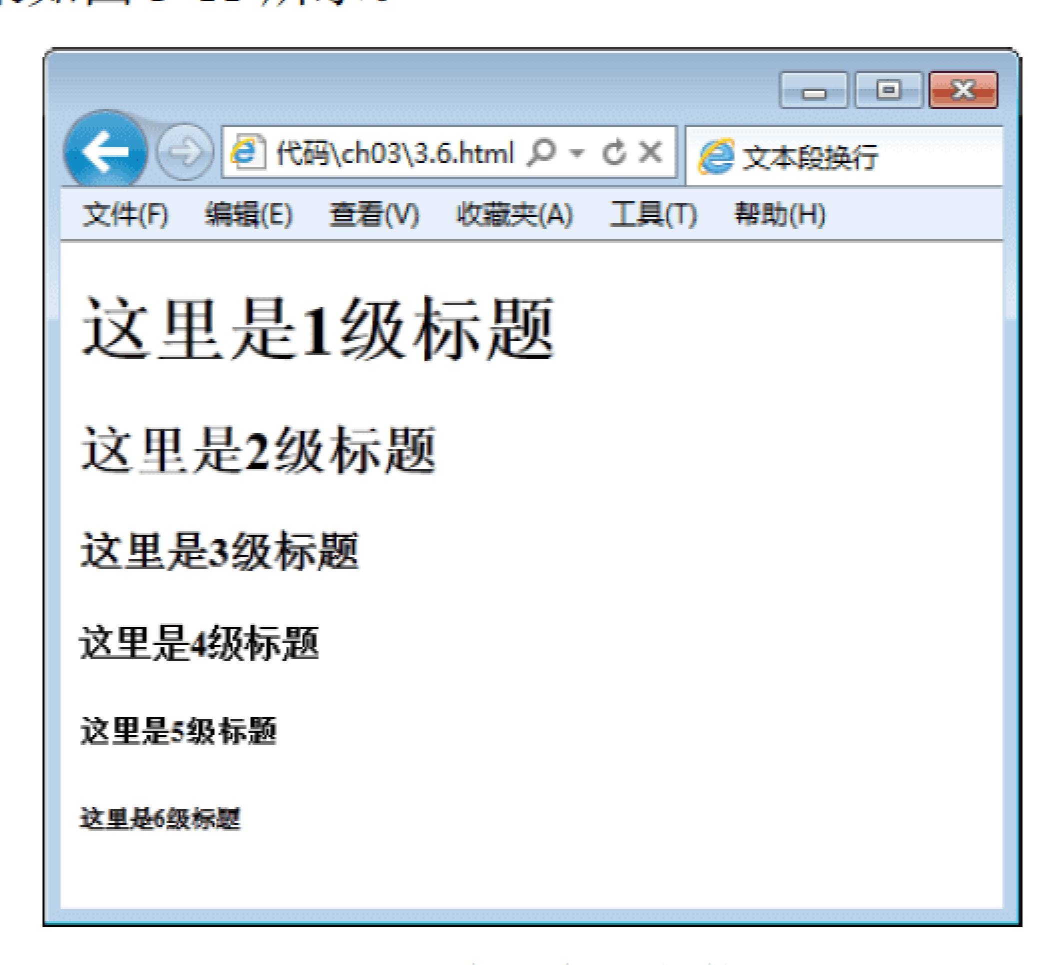


图 3-11 标题标记的使用



作为标题，它们的重要性是有区别的，其中<h1>标题的重要性最高，<h6>重要性最低。

3.2.4　文本水平居中标记<center>

使用<center>标签，可以将其所包括的文本进行水平居中。Center 标签是成对出现的，以<center>开始，</center>结束。

【例 3.12】（实例文件：ch03\3.12.html）

```
<html>
<head>
<title>文本水平居中</title>
</head>
<body>
清明时节雨纷纷
<center>清明时节雨纷纷</center>
</body>
</html>
```

在 IE 9.0 中预览效果如图 3-12 所示。

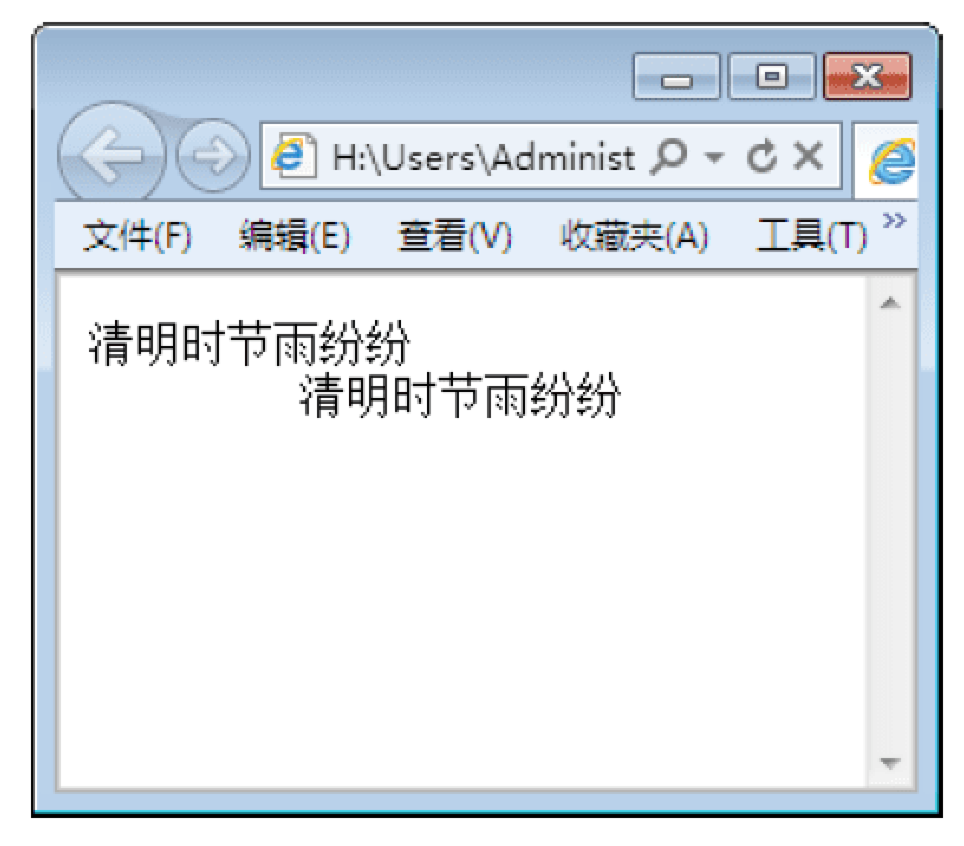


图 3-12　文本水平居中标记的使用



在 HTML 中，不赞成使用 center 元素。而且在 HTML 中该标记已经被停用，当需要将文字进行水平居中显示时，请使用 CSS 样式来居中文本。

3.2.5　标尺标记<hr>

在网页文本排版的过程中，如果在文本中添加标尺，可以让文本有条理地显示，尤其是当文本段落比较长时。

【例 3.13】（实例文件：ch03\3.13.html）

```
<html>
<body>
<p>定义水平线</p>
```

```
<hr />
<p>床前明月光，疑是地上霜。</p>
<hr />
<p>举头望明月，低头思故乡。</p>
</body>
</html>
```

在 IE 9.0 中预览效果如图 3-13 所示。

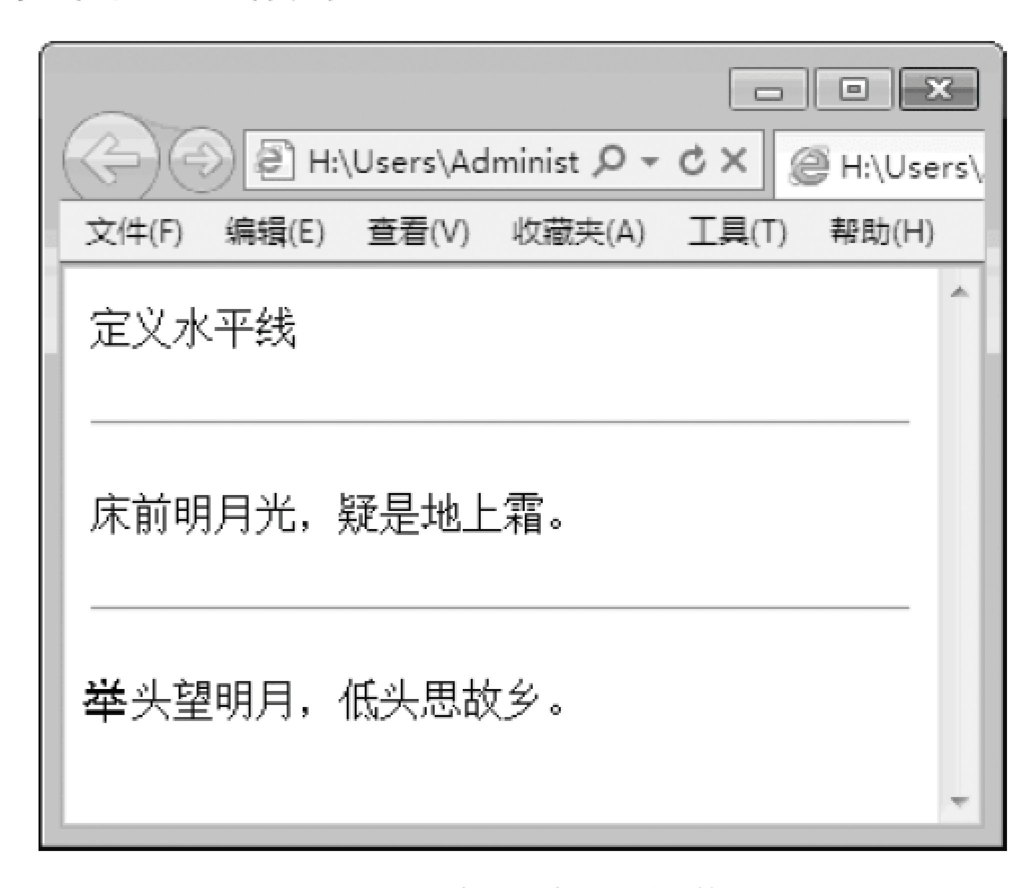


图 3-13　标尺标记的作用

3.3　网页文字列表的设计

文字列表可以有序地编排一些信息，使其结构化和条理化，并以列表的样式显示出来，以便浏览者能更加快捷地获得相应信息。HTML 中的文字列表类似于文字编辑软件 Word 中的项目符号和自动编号。

3.3.1　建立无序列表

无序列表相当于 Word 中的项目符号，无序列表的项目排列没有顺序，只以符号作为分项标识。无序列表使用
标记，其中每一个列表项使用
标记，其结构如下所示。

```
<ul>
  <li>无序列表项</li>
  <li>无序列表项</li>
  <li>无序列表项</li>
  <li>无序列表项</li>
</ul>
```

在无序列表结构中，使用
标记表示这个无序列表的开始和结束，<li>则表示一个列表项的开始。在一个无序列表中可以包含多个列表项，并且<li>可以省略结束标记。下面的实例使用无序列表实现文本的显示。

【例 3.14】（实例文件：ch03\3.14.html）

```
<html>
<head>
<title>嵌套无序列表的使用</title>
</head>
<body>
<h1>网站建设流程</h1>
<ul>
    <li>项目需求</li>
    <li>系统分析
      <ul>
        <li>网站的定位</li>
        <li>内容收集</li>
        <li>栏目规划</li>
        <li>网站目录结构设计</li>
        <li>网站标志设计</li>
        <li>网站风格设计</li>
        <li>网站导航系统设计</li>
      </ul>
   </li>
   <li>伪网页草图
      <ul>
        <li>制作网页草图</li>
        <li>将草图转换为网页</li>
      </ul>
   </li>
   <li>站点建设</li>
   <li>网页布局</li>
   <li>网站测试</li>
   <li>站点的发布与站点管理</li>
</ul>
</body>
</html>
```

在 IE 9.0 中预览效果如图 3-14 所示。读者会发现，无序列表项中可以嵌套一个列表。如代码中的“系统分析”列表项和“伪网页草图”列表项中都有下级列表，因此在这对
标记之间又增加了一对
标记。

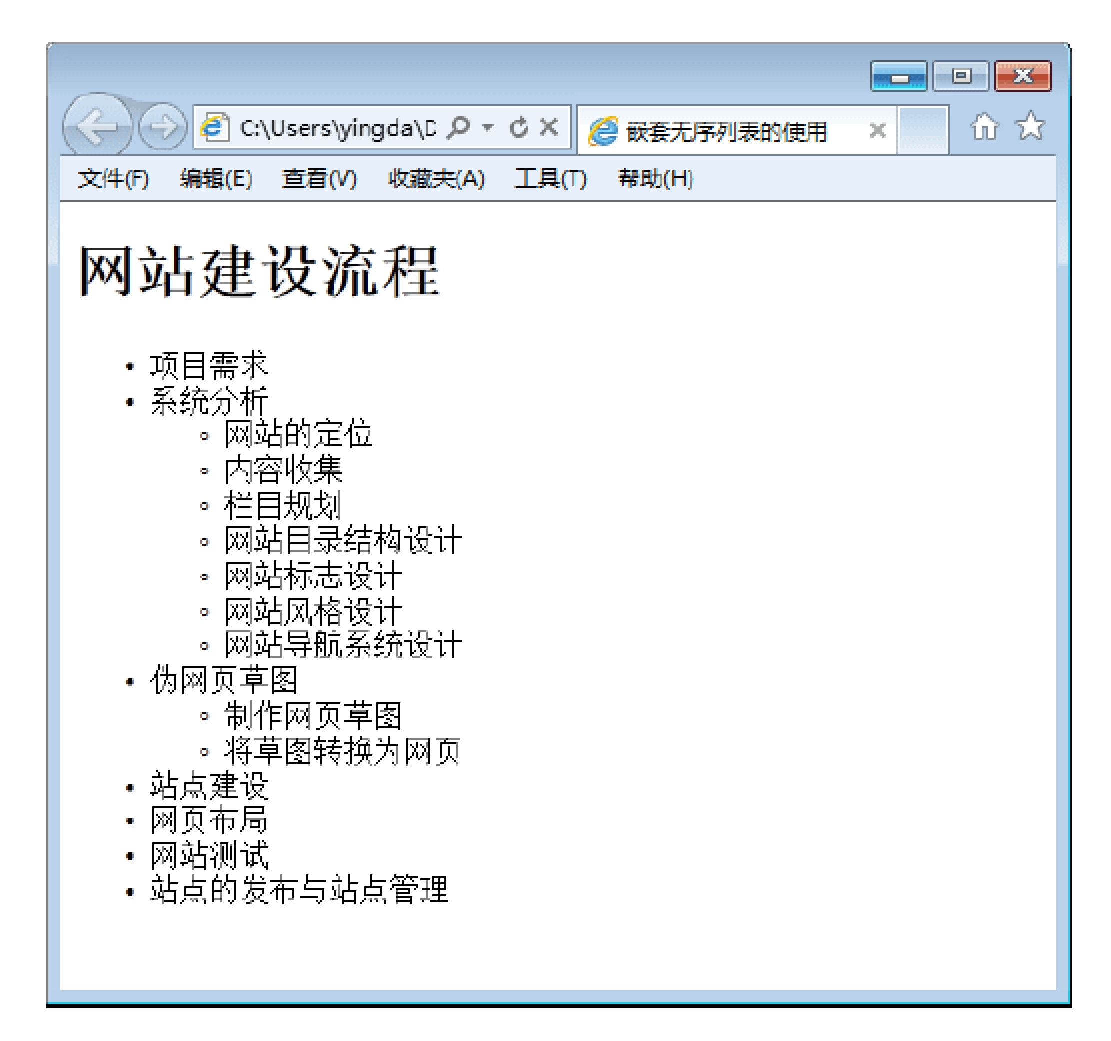


图 3-14　无序列表

3.3.2　建立有序列表

有序列表类似于 Word 中的自动编号功能，有序列表的使用方法和无序列表的使用方法基本相同，它使用标记
，每一个列表项前使用
。每个项目都有前后顺序之分，多数用数字表示，其结构如下。

```
<ol>
  <li>第 1 项</li>
  <li>第 2 项</li>
  <li>第 3 项</li>
</ol>
```

下面的实例使用有序列表实现文本的排列显示。

【例 3.15】（实例文件：ch03\3.15.html）

```
<html>
<head>
<title>有序列表的使用</title>
</head>
<body>
<h1>本讲目标</h1>
<ol>
  <li>网页的相关概念 </li>
  <li>网页与 HTML</li>
  <li> Web 标准（结构、表现、行为）</li>
  <li>网页设计与开发的过程 </li>
  <li>与设计相关的技术因素</li>
  <li>HTML 简介 </li>
```

```
</ol>
</body>
</html>
```

在 IE 9.0 中预览效果如图 3-15 所示。用户可以看到新添加的有序列表。

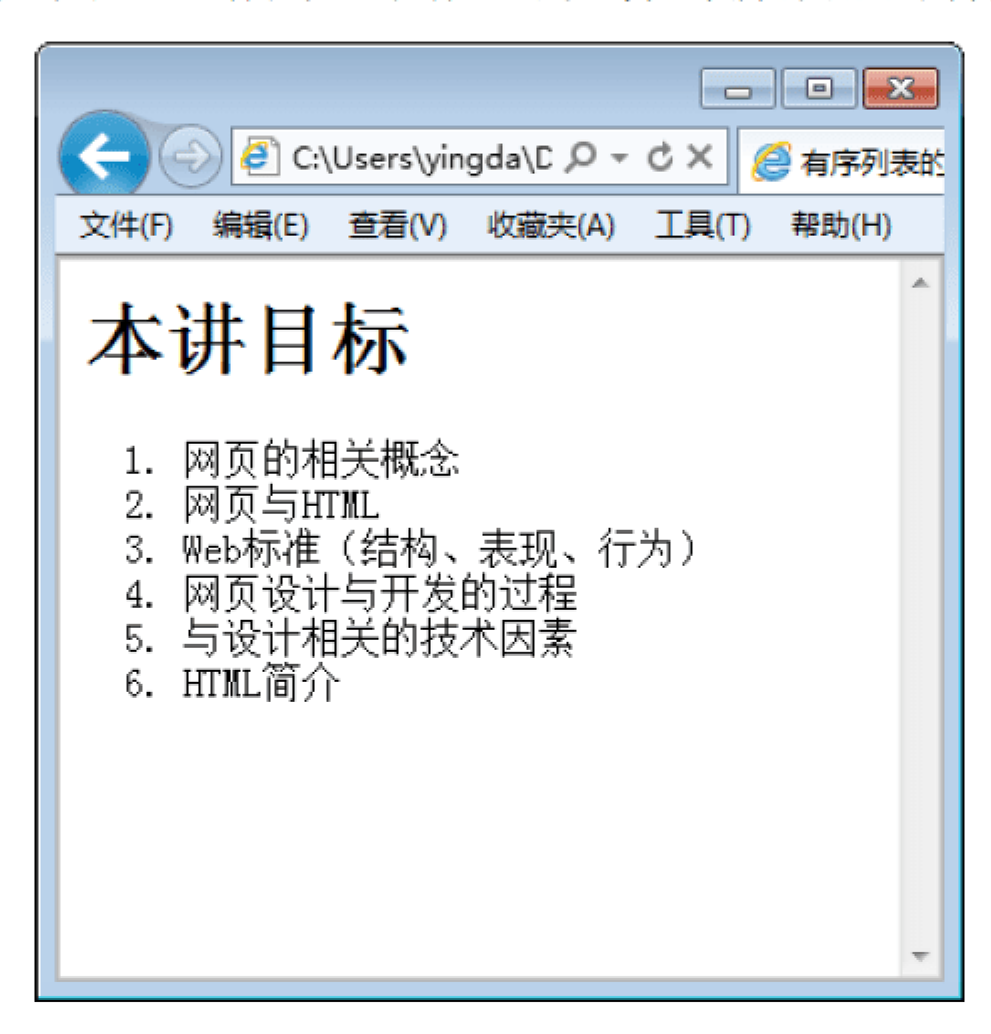


图 3-15　有序列表的效果

3.3.3　建立不同类型的无序列表

通过使用多个<ul>标签，可以建立不同类型的无序列表。

【例 3.16】（实例文件：ch03\3.16.html）

```
<html>
<body>
<h4>Disc 项目符号列表：</h4>
<ul type="disc">
 <li>苹果</li>
 <li>香蕉</li>
 <li>柠檬</li>
 <li>桔子</li>
</ul>
<h4>Circle 项目符号列表：</h4>
<ul type="circle">
 <li>苹果</li>
 <li>香蕉</li>
 <li>柠檬</li>
 <li>桔子</li>
</ul>
<h4>Square 项目符号列表：</h4>
```

```
<ul type="square">
 <li>苹果</li>
 <li>香蕉</li>
 <li>柠檬</li>
 <li>桔子</li>
</ul>
</body>
</html>
```

在 IE 9.0 中预览效果如图 3-16 所示。

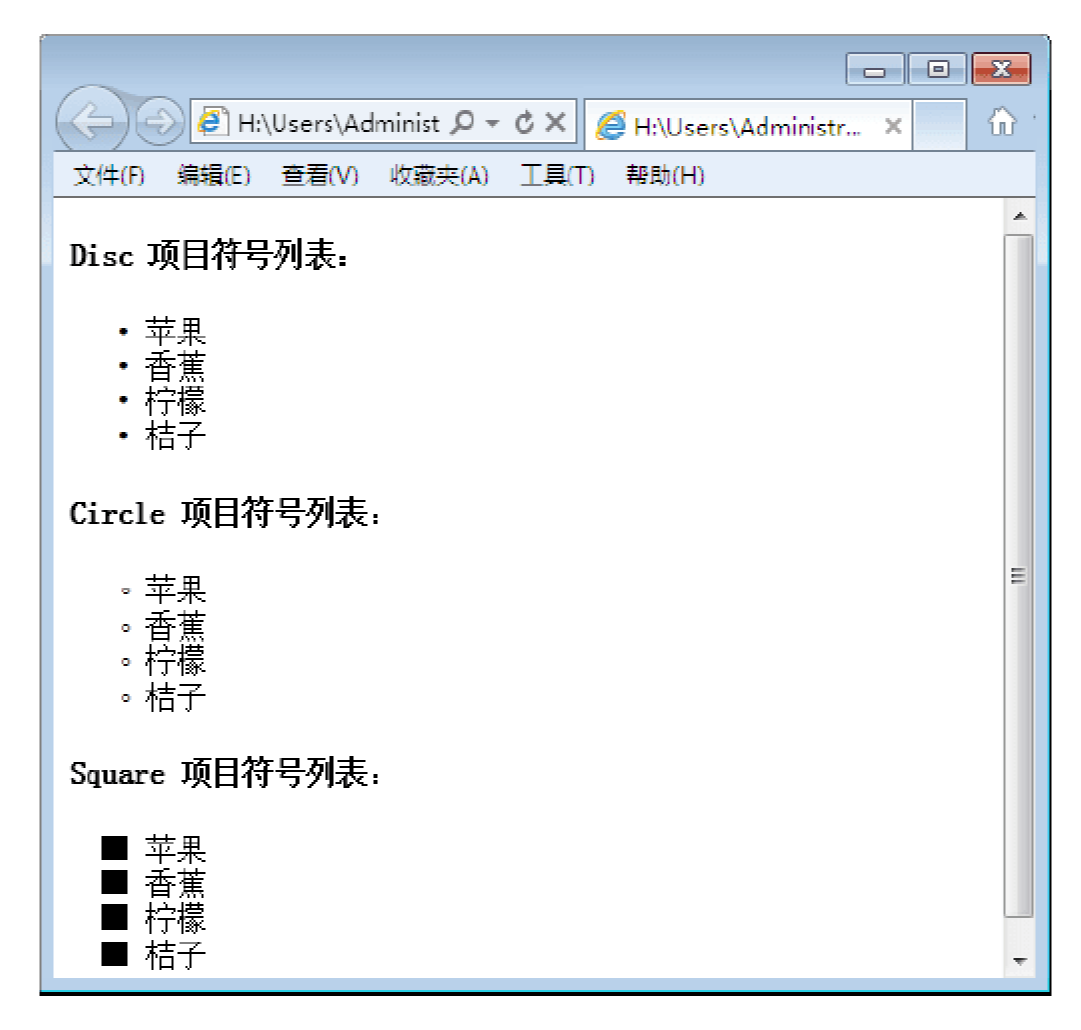


图 3-16　不同类型的无序列表

3.3.4　建立不同类型的有序列表

通过使用多个<ol>标签，可以建立不同类型的有序列表。

【例 3.17】（实例文件：ch03\3.17.html）

```
<html>
<body>
<h4>数字列表：</h4>
<ol>
 <li>苹果</li>
 <li>香蕉</li>
 <li>柠檬</li>
 <li>桔子</li>
</ol>
<h4>字母列表：</h4>
<ol type="A">
 <li>苹果</li>
```

```
  <li>香蕉</li>
  <li>柠檬</li>
  <li>桔子</li>
</ol>
</body>
</html>
```

在 IE 9.0 中预览效果如图 3-17 所示。

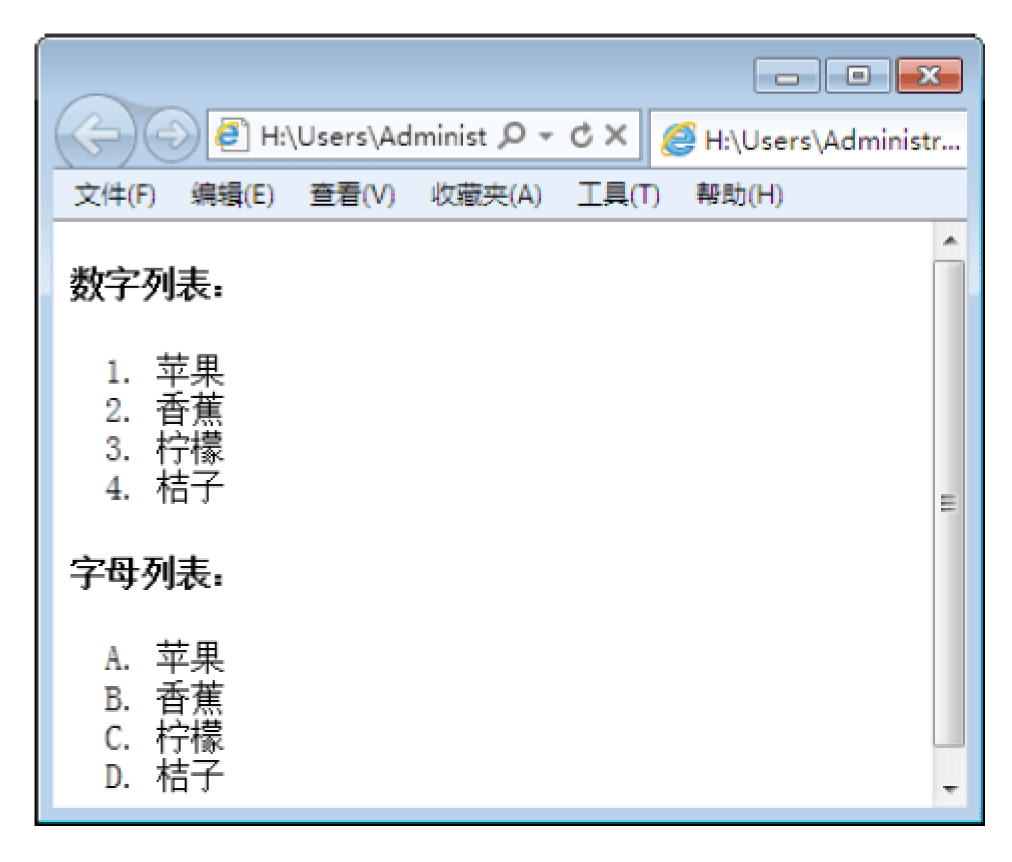


图 3-17　不同类型的有序列表

3.3.5　嵌套列表

嵌套列表是网页中常用的元素，使用<ul>标签可以制作网页中的嵌套列表。

【例 3.18】（实例文件：ch03\3.18.html）

```
<html>
<body>
<h4>一个嵌套列表：</h4>
<ul>
  <li>咖啡</li>
  <li>茶
    <ul>
    <li>红茶</li>
    <li>绿茶
      <ul>
      <li>中国茶</li>
      <li>非洲茶</li>
      </ul>
    </li>
    </ul>
  </li>
  <li>牛奶</li>
```

```
</ul>
</body>
</html>
```

在 IE 9.0 中预览效果如图 3-18 所示。

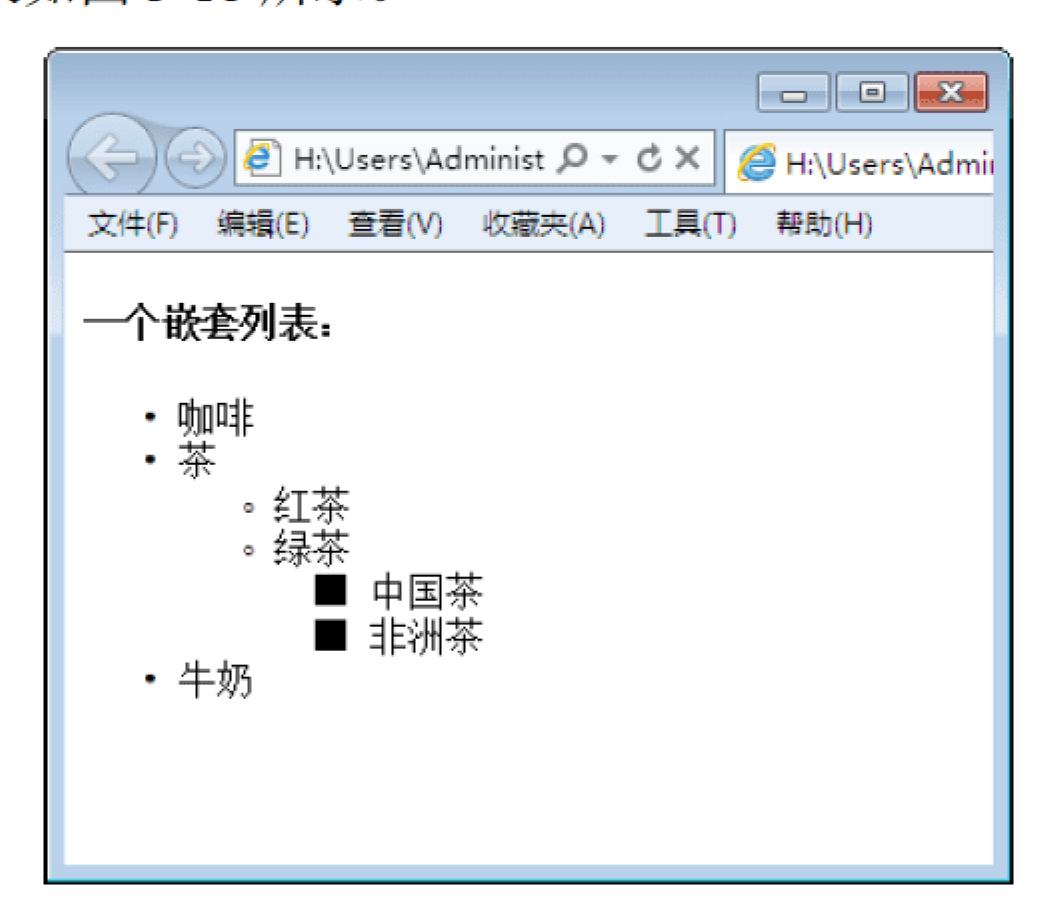


图 3-18 嵌套列表

3.3.6 自定义列表

在 HTML 中还可以自定义列表。自定义列表的标签是<dl>。

【例 3.19】（实例文件：ch03\3.19.html）

```
<html>
<body>
<h2>一个定义列表：</h2>
<dl>
    <dt>电脑</dt>
    <dd>是一种能够按照程序运行的电子设备……</dd>
    <dt>显示器</dt>
    <dd>以视觉方式显示信息的装置……</dd>
</dl>
</body>
</html>
```

在 IE 9.0 中预览效果如图 3-19 所示。

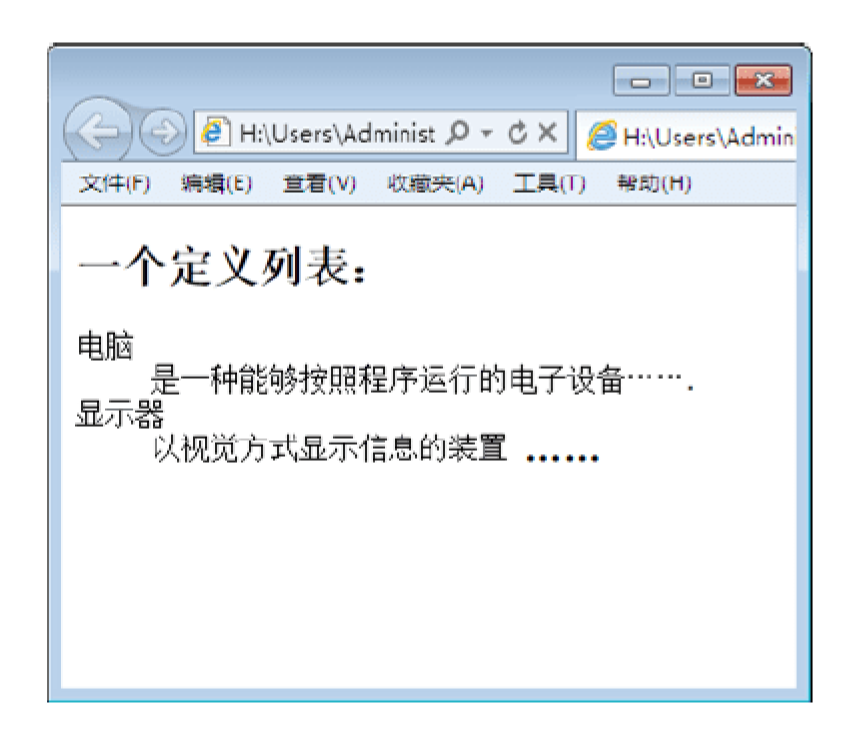


图 3-19　自定义列表

3.4　设置网页文本段落格式

段落（Paragraph）标记把文本格式化成普通段落。段落<p>标记的一个基本属性是 align（对齐），可以有 left（左）、center（中间）、right（右）几个值，默认值是 left。

3.4.1　段落左对齐

段落左对齐是段落的默认对齐方式，使用段落标记的属性 align 可以设置左对齐方式。

【例 3.20】（实例文件：ch03\3.20.html）

```
<html>
  <head>
    <title>左对齐段落</title>
  </head>
  <body>
  朝辞白帝彩云间，千里江陵一日还。
  <p align="left">两岸猿声啼不住，轻舟已过万重山。
</body>
</html>
```

在 IE 9.0 中预览效果如图 3-20 所示。

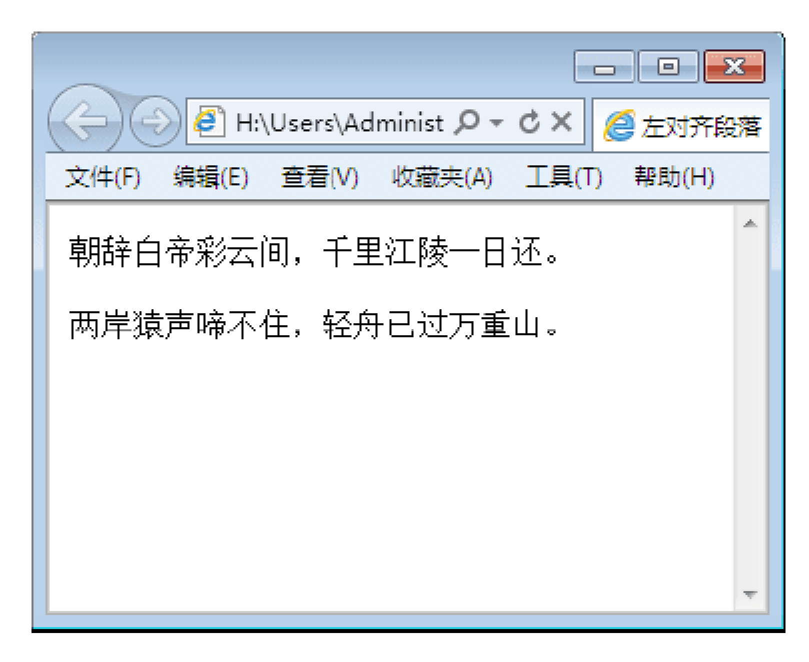


图 3-20　左对齐

3.4.2 段落居中对齐

使用段落标记的 align 属性可以设置段落的居中对齐方式。

【例 3.21】（实例文件：ch03\3.21.html）

```
<html>
  <head>
    <title>居中对齐段落</title>
  </head>
  <body>
   朝辞白帝彩云间，千里江陵一日还。
  <p align="center">两岸猿声啼不住，轻舟已过万重山。
</body>
</html>
```

在 IE 9.0 中预览效果如图 3-21 所示。

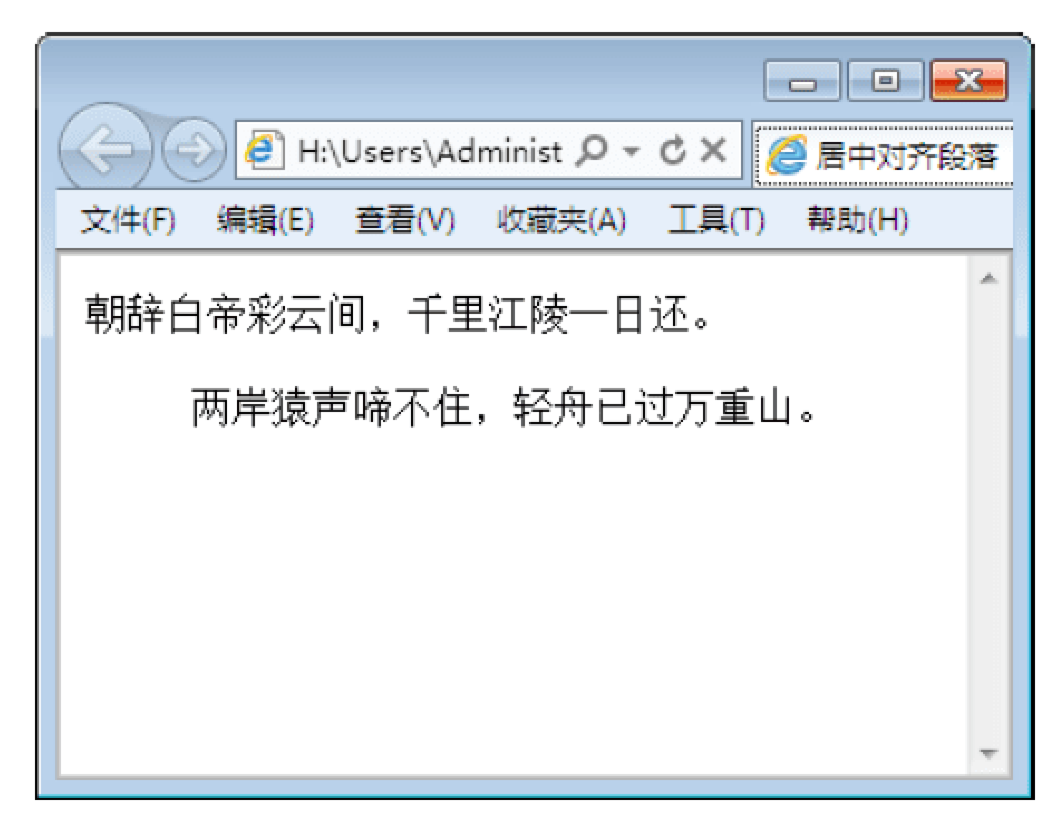


图 3-21 居中对齐

3.4.3 段落右对齐

使用段落标记的 align 属性可以设置段落的右对齐方式。

【例 3.22】（实例文件：ch03\3.22.html）

```
<html>
  <head>
    <title>居中对齐段落</title>
  </head>
  <body>
   朝辞白帝彩云间，千里江陵一日还。
  <p align="right">两岸猿声啼不住，轻舟已过万重山。
</body>
</html>
```

在 IE 9.0 中预览效果如图 3-22 所示。

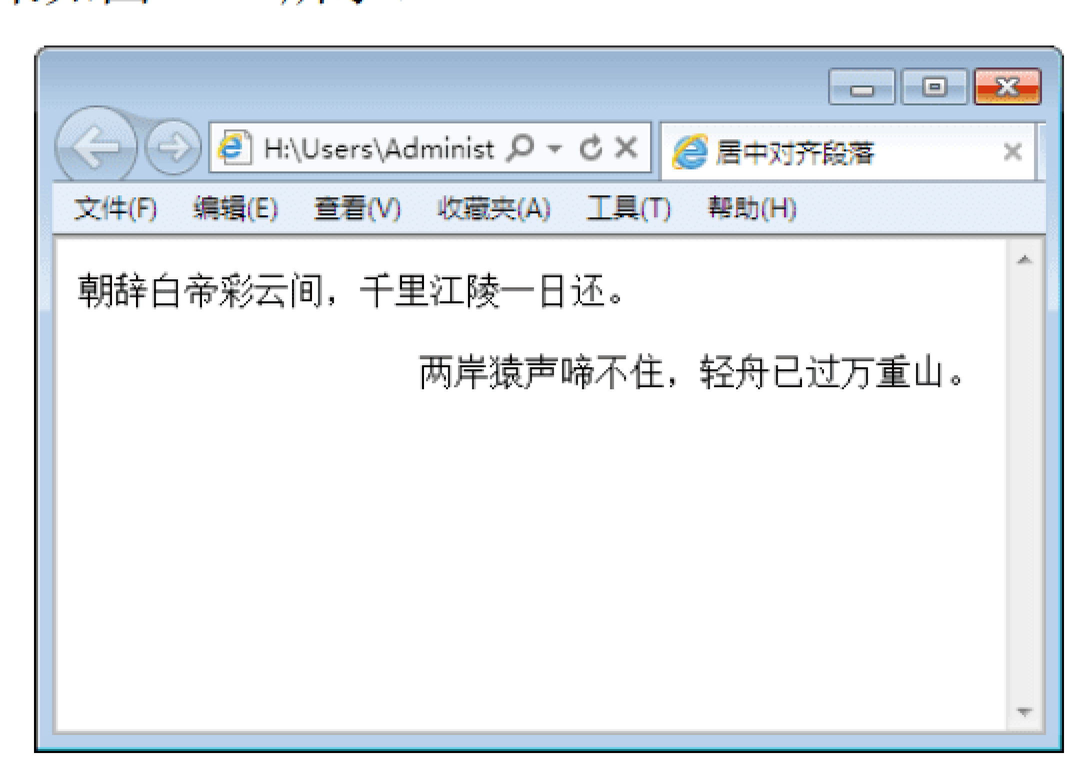


图 3-22　右对齐

3.5　综合实战——成才教育网文本设计

本章讲述了网页组成元素中最常用的文本。本实例将综合运用网页文本的设计方法，制作成才教育网的文本页面。具体操作步骤如下。

01 新建 HTML 文档，代码如下。

```
<html >
<head>
<meta charset="utf-8" />
<title>成才教育网</title>
</head>
<body>
</body>
</html>
```

02 在 body 部分增加如下 HTML 代码。保存页面。

```
<p><h2>成才教育</h2></p>
<p>成才教育成立于 2003 年，是一家专业致力于学生学习能力开发和 培养、学习社区建设、课外辅导服务、家庭教育研究的新型综合教育服务机构。自成立起，一直专业致力于初高中学生的课外辅导和学习能力的培养。</p>
<h3>教学模式</h3>
<p>为学生量身定制最佳的学习方案，改善学习方法，充分挖掘学生们的智力潜能，激发学习兴趣，培养学生的自学能力，辅导老师（以一线重点在校教师为主）对学生设计适合学生的辅导教案与作业习题</p>
<h3>教学特色</h3>
    分析学科不足制定辅导计划；<br />
    特级名师高考难点点睛；<br />
    专人陪读随时解除疑难；<br />
```

```
专业学科导师一对一面授学科知识、解题技巧、学习方式。
```

03 使用 IE 9.0 打开文件，预览效果如图 3-23 所示。

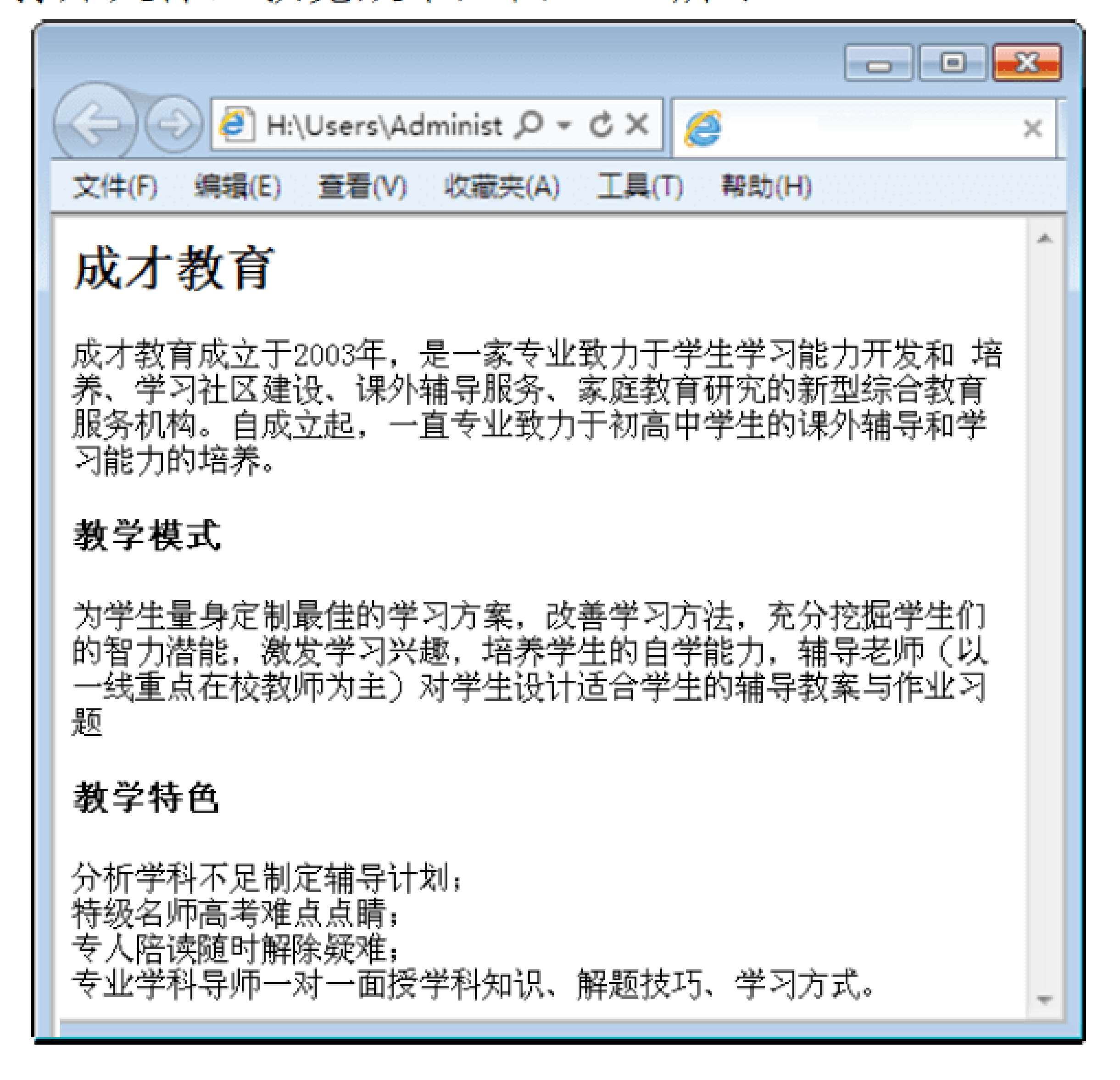


图 3-23 IE 9.0 浏览效果

3.6 专家解惑

1. 换行标记和段落标记的区别是什么？

答：换行标记是单标记，不能写结束标记。段落标记是双标记，可以省略结束标记也可以不省略。默认情况下，段落之间的距离和段落内部的行间距是不同的，段落间距比较大，行间距比较小。HTML 无法调整段落间距和行间距，如果希望调整它们，就必须使用 CSS。在 Dreamweaver CS5.5 的设计视图下，按下回车（Enter）键可以快速换段，按下 Shift+回车（Enter）键可以快速换行。

2. 无序列表<ul>元素的作用有哪些？

答：无序列表元素主要用于条理化和结构化文本信息。在实际开发中，无序列表在制作导航菜单时使用广泛。导航菜单的结构一般都使用无序列表实现。

第 4 章

网页色彩和图片的设计

色彩与图片在网站设计中占据着相当重要的地位，无论是静态网页还是动态网页，色彩与图片永远是最重要的一环。当距离显示屏较远的时候，我们看到的不是优美的版式，而是美丽的图片与网页的色彩。

4.1 网页色彩的应用

在任何一个设计中，色彩对视觉的刺激都起到信息传达的首要作用。网页中的色彩设计是最直接的视觉效果，不同的颜色运用会给人以不同的感受，高明的设计师会运用颜色来表现网站的理念和内在品质。为了能更好地应用色彩来设计网页，下面先来了解一下色彩的基础知识。

自然界中的色彩五颜六色、千变万化，比如玫瑰是红色的、大海是蓝色的、橘子是橙色的……但是最基本的色彩只有 3 种（红、黄、蓝），其他的色彩都可以由这 3 种调和而成，我们称这 3 种色彩为“三原色”，如图 4-1 所示。

大家平时所看到的白色光经过分析，在色带上可以看到，它包括红、橙、黄、绿、青、蓝、紫 7 种颜色，各颜色间自然过渡，其中红、黄、蓝是三原色，通过将三原色以不同比例混合可以得到各种颜色，如图 4-2 所示。

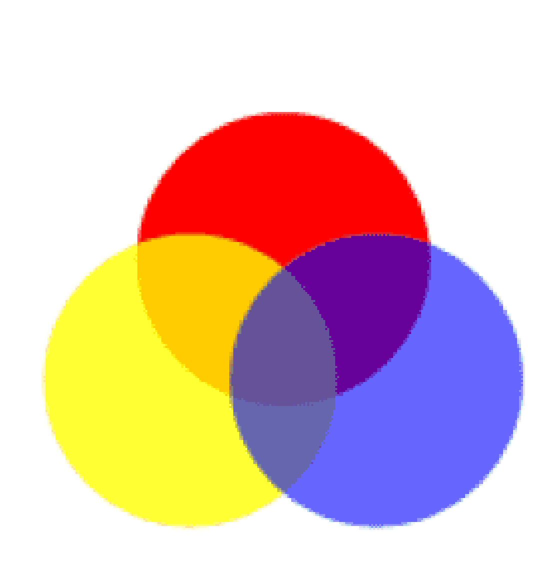


图 4-1　三原色

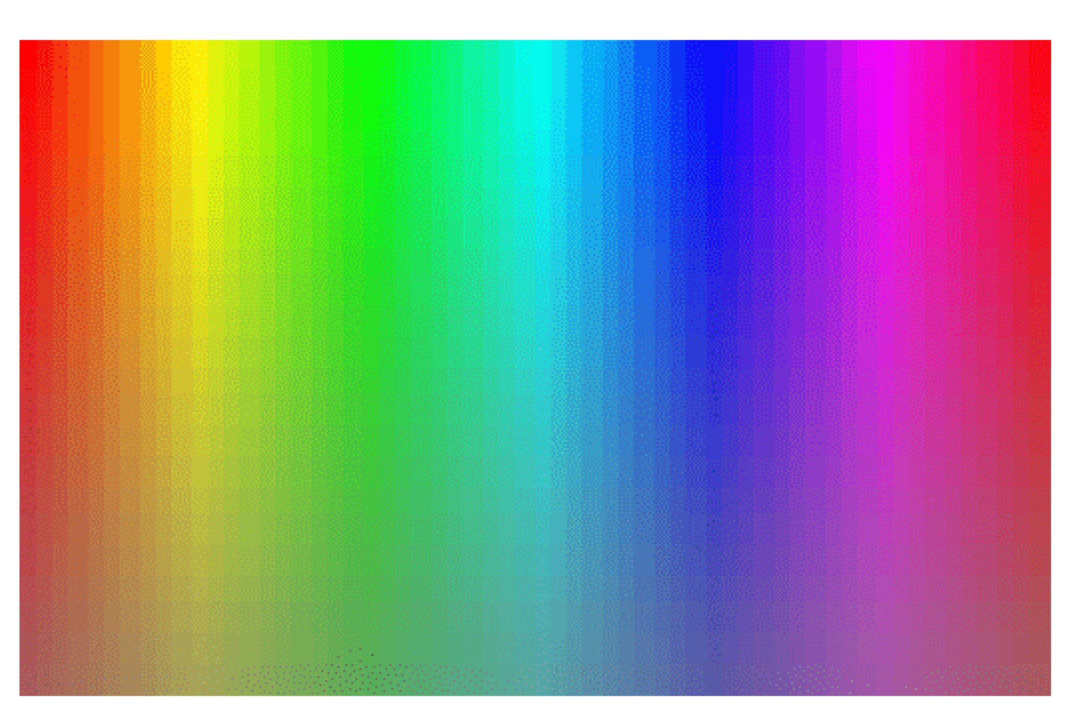


图 4-2　颜色色块

现实生活中的色彩可以分为彩色和非彩色两种，其中黑、白、灰属于非彩色系列，其他的色彩都属于彩色系列。任何一种彩色都具备 3 个特征：色相、明度和饱和度，而非彩色只有明

度属性。



色相指的是色彩的名称，这是色彩最基本的特征，反映颜色的基本面貌，是一种色彩区别于另一种色彩的最主要因素，比如紫色、绿色、黄色等都代表了不同的色相。

同一色相的色彩调整一下亮度或纯度很容易搭配，比如深绿、暗绿、草绿、亮绿。



明度也叫亮度，指的是色彩的明暗程度，明度越大，色彩越亮。

比如一些购物、儿童类网站用的是一些鲜亮的颜色，让人感觉绚丽多姿、生气勃勃，明度越低，颜色越暗，如图 4-3 所示。较暗的明度主要用于一些游戏类网站，可使网站充满神秘感，如图 4-4 所示。



图 4-3 儿童类网站



图 4-4 游戏类网站



饱和度也叫纯度，指色彩的鲜艳程度。饱和度高的色彩纯、鲜亮，饱和度低的色彩暗淡，含灰色。

非彩色只有明度属性，没有色相和饱和度属性。制作网页时使用彩色还是非彩色好呢？专业的研究机构表明：彩色的记忆效果是黑白的 3.5 倍。也就是说，在一般情况下，彩色页面较完全黑白页面更加吸引人。进行网页制作时，通常是将主要内容（如文字）用非彩色（黑色），边框、背景、图片用彩色。这样页面整体不单调，显得和谐统一。

4.2 在网页中插入图像

图像可以美化网页，插入图像使用单标记<img>。img 标记的属性及描述如表 4-1 所示。

表 4-1　img 标记的属性

属性	值	描述
alt	text	定义有关图形的短的描述
src	URL	要显示的图像的 URL
height	pixels %	定义图像的高度
ismap	URL	把图像定义为服务器端的图像映射
usemap	URL	定义作为客户端图像映射的一幅图像。请参阅 <map> 和 <area> 标签，了解其工作原理
vspace	pixels	定义图像顶部和底部的空白。不支持。请使用 CSS 代替
width	pixels %	设置图像的宽度

1．插入图像

src 属性用于指定图片源文件的路径，它是 img 标记必不可少的属性。其语法格式如下。

```
<img src="图片路径">
```

图片的路径可以是绝对路径，也可以是相对路径。下面的实例是在网页中插入图片。

【例 4.1】（实例文件：ch04\4.1.html）

```
<html>
<head>
<title>插入图片</title>
</head>
<body>
<img src="images/meishi.jpg">
</body>
</html>
```

在 IE 9.0 中预览效果如图 4-5 所示。



图 4-5　插入图片

2. 从不同位置插入图像

在插入图片时，用户可以将其他文件夹或服务器中的图片显示到网页中。

【例 4.2】（实例文件：ch04\4.2.html）

```
<html>
<body>
<p>
来自一个文件夹的图像：
<img src="images/meishi.jpg" />
</p>
<p>
来自 baidu 的图像：
<img src="http://www.baidu.com/img/shouye_b5486898c692066bd2cbaeda86d74448.gif" />
</p>
</body>
</html>
```

在 IE 9.0 中预览效果如图 4-6 所示。



图 4-6　插入其他文件夹或服务器的图片

4.2.1　设置图像在网页中的宽度和高度

在 HTML 文档中，还可以设置插入图片的显示大小，一般是按原始尺寸显示，但也可以设置显示尺寸。设置图像尺寸时要使用属性 width（宽度）和 height（高度）。

【例 4.3】（实例文件：ch04\4.3.html）

```
<html>
<head>
<title>插入图片</title>
</head>
```

```
<body>
<img src="images/meishi.jpg">
<img src="images/meishi.jpg" width="200">
<img src="images/meishi.jpg" width="200" height="300">
</body>
</html
```

在 IE 9.0 中预览效果，如图 4-7 所示。

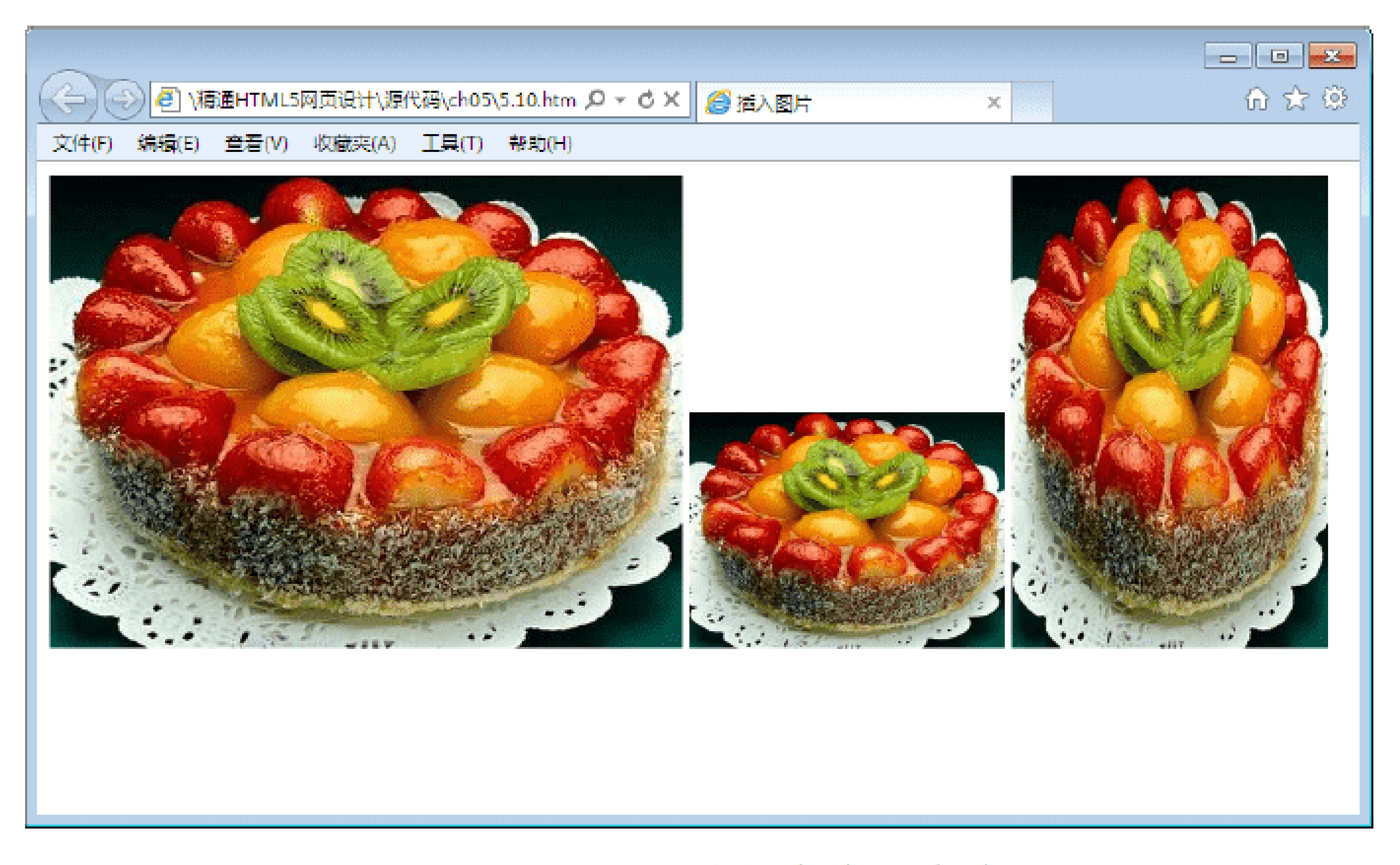


图 4-7　设置图片的宽度和高度

由图 4-7 可以看到，图片的显示尺寸由 width（宽度）和 height（高度）控制。当只为图片设置一个尺寸属性时，另外一个尺寸就以图片原始的长宽比例来显示。图片的尺寸单位可以选择百分比或数值。百分比为相对尺寸，数值是绝对尺寸。



如果网页中插入的图像都是位图，放大尺寸，图像会出现马赛克，变得模糊。



在 Windows 中查看图片的尺寸，只需要找到图像文件，把鼠标指针移动到图像上，停留几秒后，就会出现一个提示框，说明图像文件的尺寸。尺寸后显示的数字代表图像的宽度和高度，如 256 × 256。

4.2.2　设置图像的提示文字

图像提示文字的作用有两个。首先，当浏览网页时，如果图像下载完成，将鼠标指针放在该图像上，鼠标指针旁边会出现提示文字。其次，如果图像没有成功下载，在图像的位置上就会显示提示文字。设置提示文字的属性为 alt。

随着互联网技术的发展，网速已经不是制约因素，因此一般都能成功下载图像。现在，在百度、google 等大型搜索引擎中，搜索图片不如文字方便，如果给图片添加适当提示，可以方便搜索引擎的检索。

下面的实例为图片添加提示文字效果。

【例 4.4】（实例文件：ch04\4.4.html）

```
<html>
<head>
<title>图片文字提示</title>
</head>
<body>
<img src="images/meishi.jpg" alt="草莓甜橙沙拉">
</body>
</html>
```

在 IE 9.0 中预览效果如图 4-8 所示。用户将鼠标放在图片上，即可看到提示文字。



图 4-8　图片文字提示



在火狐浏览器中不支持该功能。

4.2.3　将图片设置为网页背景

在插入图片时，用户可以根据需要将某些图片设置为网页的背景。Gif 和 jpg 文件均可用作 HTML 网页的背景。如果图像小于页面大小，图像会重复显示。

【例 4.5】（实例文件：ch04\4.5.html）

```
<html>
<body background="images/background.jpg">
<h3>图像背景</h3>
</body>
</html>
```

在 IE 9.0 中预览效果如图 4-9 所示。



图 4-9　图片背景

4.2.4　排列图像

如果在网页的文字中插入图片，就可以对图像进行排列。常用的排列方式为居中、底部对齐、顶部对齐。

【例 4.6】（实例文件：ch04\4.6.html）

```
<html>
<body>
<h2>未设置对齐方式的图像：</h2>
<p>图像<img src ="images/logo.gif"> 在文本中</p>
<h2>已设置对齐方式的图像：</h2>
<p>图像 <img src=" images/logo.gif " align="bottom"> 在文本中</p>
<p>图像 <img src =" images/logo.gif " align="middle"> 在文本中</p>
<p>图像 <img src =" images/logo.gif " align="top"> 在文本中</p>
</body>
</html>
```

在 IE 9.0 中预览效果如图 4-10 所示。



底部对齐方式是默认的对齐方式。

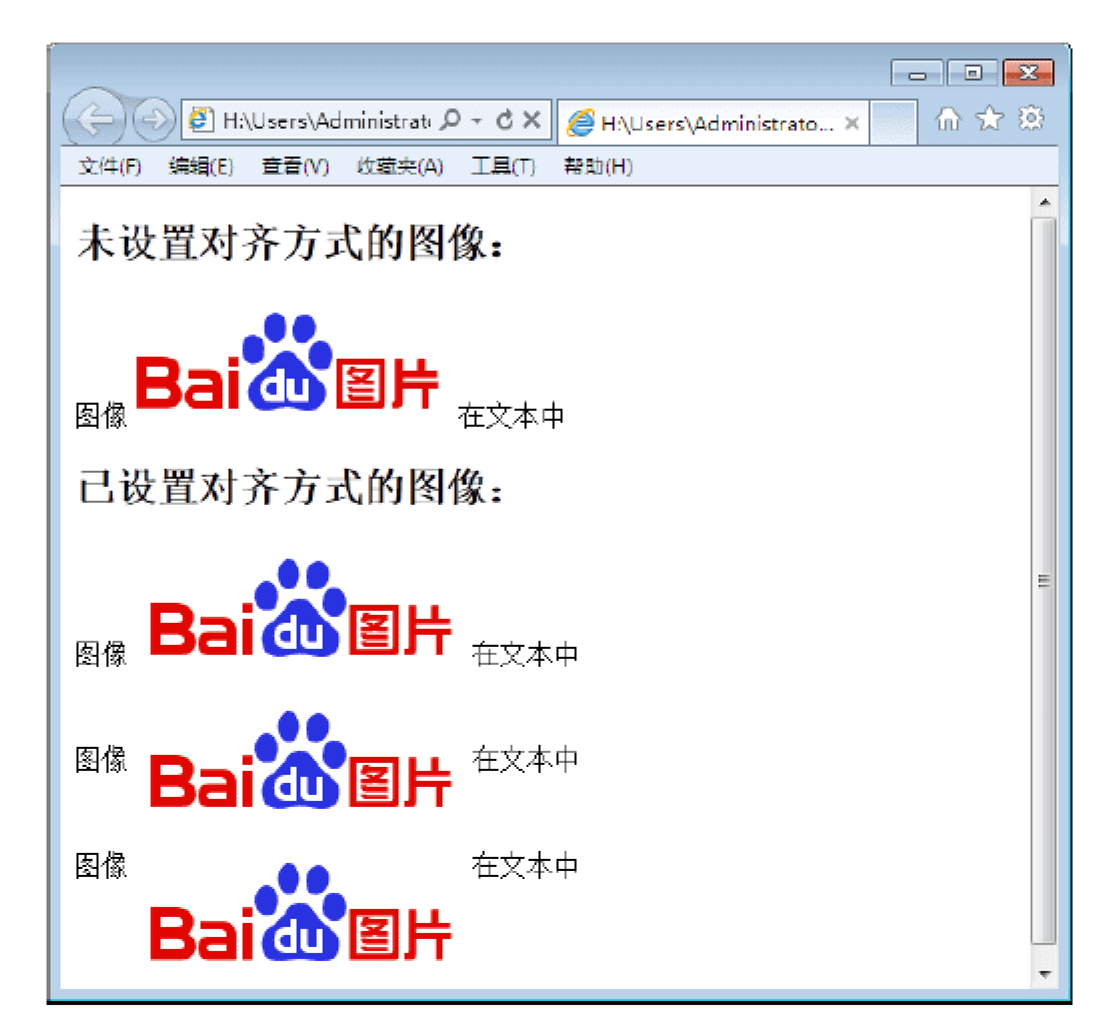


图 4-10 图片对齐方式

4.3 综合实战——制作茂森房地产广告网页

本章讲述了网页组成元素中最常用的文本和图片。下面的综合实例创建了一个由文本和图片构成的房屋装饰效果网页，如图 4-11 所示。



图 4-11 房屋装饰效果网页

具体操作步骤如下。

01 在 Dreamweaver CS5.5 中新建 HTML 文档，并修改成 HTML 标准，代码如下。

```
<html >
<head>
```

```
<title>房屋装饰装修效果图</title>
</head>
<body>
</body>
</html>
```

02 在 body 部分增加如下 HTML 代码。保存页面。

```
<p> <img src="images/xiyatu.jpg" width="300" height="200"/> <img src="images/stadshem.jpg" width="300" height="200"/><br />
西雅图原生态公寓室内设计与 Stadshem 小户型公寓设计（带阁楼）</p>
<hr/>
<p> <img src="images/qingxinhuoli.jpg" width="300" height="200"/> <img src="images/renwen.jpg" width="300" height="200"/><br />
清新活力家居与人文简约悠然家居</p>
<hr />
```



<hr>标记的作用是定义内容中的主题变化，并显示为一条水平线，在 HTML 中它没有任何属性。

另外，要快速插入图片，可以借助 Dreamweaver CS5.5 的插入功能，或按下快捷键 Ctrl+Alt+I。

4.4　专家解惑

1. 在浏览器中无法显示图片。

答：图片在网页中属于嵌入对象，并不是保存在网页中，网页只是保存了指向图片的路径。浏览器在解释 HTML 文件时，会按指定的路径去寻找图片，如果在指定的位置不存在图片，图片就无法正常显示。为了保证图片的正常显示，制作网页时需要注意以下几点。

（1）图片格式一定是网页支持的。

（2）图片的路径一定要正常，并且图片文件扩展名不能省略。

（3）HTML 文件位置发生改变时，图片一定要跟着改变，即图片位置和 HTML 文件位置始终保持相对一致。

2. 网页设计中，图像如何使用？

图像内容应有一定的实际作用，切忌虚饰浮夸。图画可以弥补文字之不足，但并不能完全取代文字。很多用户把浏览软件设定为略去图像，以求节省时间，他们只看文字。因此，制作主页时，必须注意将图像所连接的重要信息或联接其他页面的指示用文字重复表达几次，同时要注意避免使用过大的图像，如果不得不放置大的图像在网站上，应该把图像的缩小版本的预览效果显示出来，这样用户就不必浪费金钱和时间去下载他们根本不想看的大图像。

第 5 章

网页超链接的设计

HTML 文件中最重要的应用之一就是超链接，超链接是一个网站的灵魂，Web 上的网页是互相链接的，单击被称为超链接的文本或图形就可以链接到其他页面。

5.1　链接和路径

链接是网页中极为重要的部分，单击文档中的链接，即可跳转至相应位置。正是有了链接，才可以在网站中相互跳转而方便地查阅各种各样的知识，享受网络带来的无穷乐趣。

5.1.1　超链接的概念

浏览网页就是从一个文档跳转到另一个文档，从一个位置跳转到另一个位置，从一个网站跳转到另一个网站的过程，而这些过程都是通过链接来实现的。

利用链接可以实现文档之间或文档中的跳转。链接由两个端点（也称锚）和一个方向构成，通常将开始位置的端点称做源端点（或源锚），而将目标位置的端点称为目标端点（或目标锚），链接就是从源端点到目标端点的一种跳转。如图 5-1 所示，在 114 啦网址导航页面中单击【百度】超级链接，即可打开百度首页，如图 5-2 所示。



图 5-1　114 啦网址导航页面



图 5-2　百度首页



目标端点可以是任意的网络资源，例如，可以是一个页面、一幅图像、一段声音、一段程序，甚至可以是页面中的某个位置。

5.1.2 链接路径 URL

URL 为 Uniform Resource Locator 的缩写，通常翻译为“统一资源定位器”，也就是人们常说的“网址”。它用于指定 Internet 上的资源位置。

1. URL 的格式

网络中的计算机之间是通过 IP 地址区分的，如果希望访问网络中某台计算机中的资源，首先要定位到这台计算机。IP 地址由 32 位的二进制，即 32 个 0/1 代码组成，数字之间没有意义，不容易记忆。为了方便记忆，现在计算机一般采用域名的方式来寻址，即在网络上使用一组有意义字符组成的地址代替 IP 地址来访问网络资源。

URL 由 4 个部分组成，即“协议”、“主机名”、“文件夹名”和“文件名”，如图 5-3 所示。

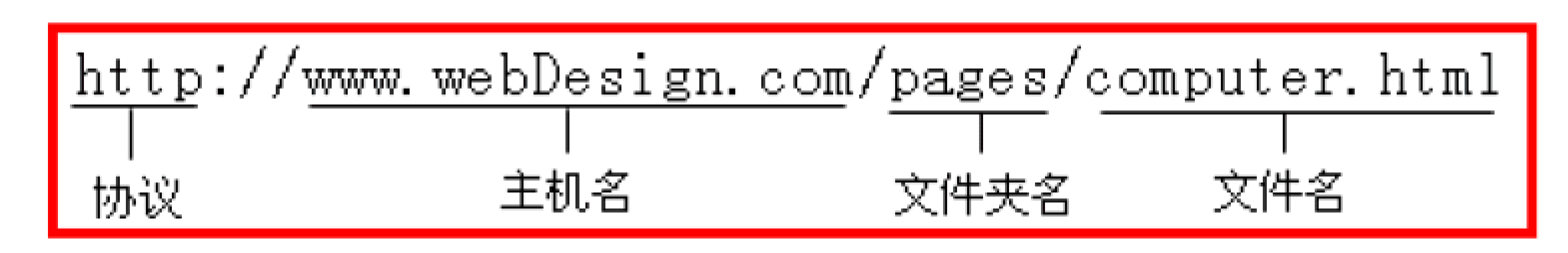


图 5-3 URL 组成

互联网中有各种各样的应用，如 Web 服务、FTP 服务等。每种服务应用都有对应的协议，通常通过浏览器浏览网页的协议都是 HTTP 协议，即“超文本传输协议”，因此通常网页的地址都以“http://”开头。“www.baidu.com”为主机名，表示文件存在于哪台服务器，主机名可以通过 IP 地址或者域名来表示。确定到主机后，还需要说明文件存在于这台服务器的哪个文件夹中，这里文件夹可以分为多个层级。确定文件夹后，就要定位到文件，即要显示哪个文件，网页文件通常是以“.html”或“.htm”为扩展名。

2. URL 的类型

超链接的 URL 可以分为两种类型：绝对 URL 和相对 URL。

- 绝对 URL 一般用于访问非同一台服务器上的资源。
- 相对 URL 是指访问同一台服务器上相同文件夹或不同文件夹中的资源。如果访问相同文件夹中的文件，只需要写文件名；如果访问不同文件夹中资源，URL 以服务器的根目录为起点，指明文档的相对关系，由文件夹名和文件名两部分构成。

下面的实例分别使用绝对 URL 和相对 URL 实现超链接。

【例 5.1】（实例文件：ch05\5.1.html）

```
<html>
<head>
```

```
<title>绝对 URL 和相对 URL</title>
</head>
<body>
  单击<a href="http://www.webDesign.com/index.html">绝对 URL</a>链接到 webDesign 网站首页<br />
  单击<a href="02.html">相同文件夹的 URL</a>链接到相同文件夹中的第 2 个页面<br />
  单击<a href="../pages/03.html">不同文件夹的 URL</a>链接到不同文件夹中的第 3 个页面
</body>
</html>
```

在上述代码中，第 1 个链接使用的是绝对 URL；第 2 个使用的是服务器相对 URL，也就是链接到文档所在服务器根目录下的 02.html；第 3 个使用的是文档相对 URL，即原文档所在文件夹的父文件夹下面的 pages 文件夹中的 03.html 文件。

在 IE 9.0 中预览网页效果如图 5-4 所示。

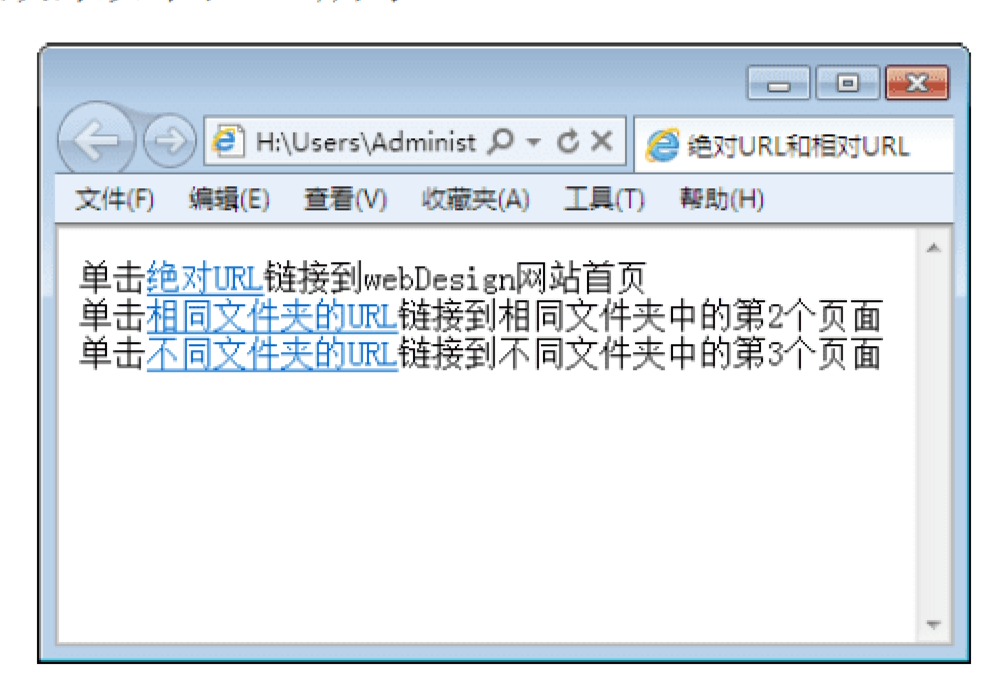


图 5-4　绝对 URL 和相对 URL

5.1.3　HTTP 路径

HTTP 路径用来链接 Web 服务器中的文档。下面是 HTTP 路径的几种写法。

- http://域名/目录/目标文件
- http://域名/目录/目标文件#片段
- http://域名/目录/目标文件？变量名=变量参数

其中第一种链接是最普通的链接格式，不使用任何参数和变量；第二种是链接到目标文件的某个一个片段的位置上；第三种是传递某个参数，使用相应的程序来处理相关内容。

【例 5.2】（实例文件：ch05\5.2.html）

```
<html>
<head>
<title>HTTP 路径</title>
</head>
<body>
链接地址：<a href="http://www.baidu.com/index.php?tn=myppcb&bar=11">链接</a>
</body>
```

```
</html>
```

在 IE 9.0 中预览网页，当单击含有链接的文本时，显示效果如图 5-5 所示。



图 5-5　使用普通 HTTP 路径的显示效果

5.1.4　FTP 路径

FTP 路径用来链接 FTP（File Transfer Protocol，文件传输协议）服务器中的文档。下面是 FTP 路径可以使用的写法。

```
<A href="ftp://……">链接文字</A>
```

其中，域名和 IP 地址是服务器地址的两种写法，都代表网络中唯一的标识。如果要在网页中使用 FTP 链接，可使用如下的写法。

```
ftp://ftp 域名/目录
```

使用 FTP 路径时，可以在浏览器中直接输入相应的 FTP 地址，打开相应的目录或者下载相关内容。也可以使用相关的软件，打开 FTP 地址中相应的目录或者下载相关内容。

【例 5.3】（实例文件：ch05\5.3.html）

```
<html>
<head>
<title>FTP 路径</title>
</head>
<body>
链接中使用 FTP 路径：<a href="ftp://ftp.lele.cn" target="_blank">链接内容</a>
</body>
</html>
```

在 IE 9.0 中预览网页，当单击含有链接的文本时，显示效果如图 5-6 所示。可以看出，此时打开一个新的窗口，显示了此目录下所有的文件，可以使用复制、粘贴功能，或者借助下载软件来下载目录中的内容。

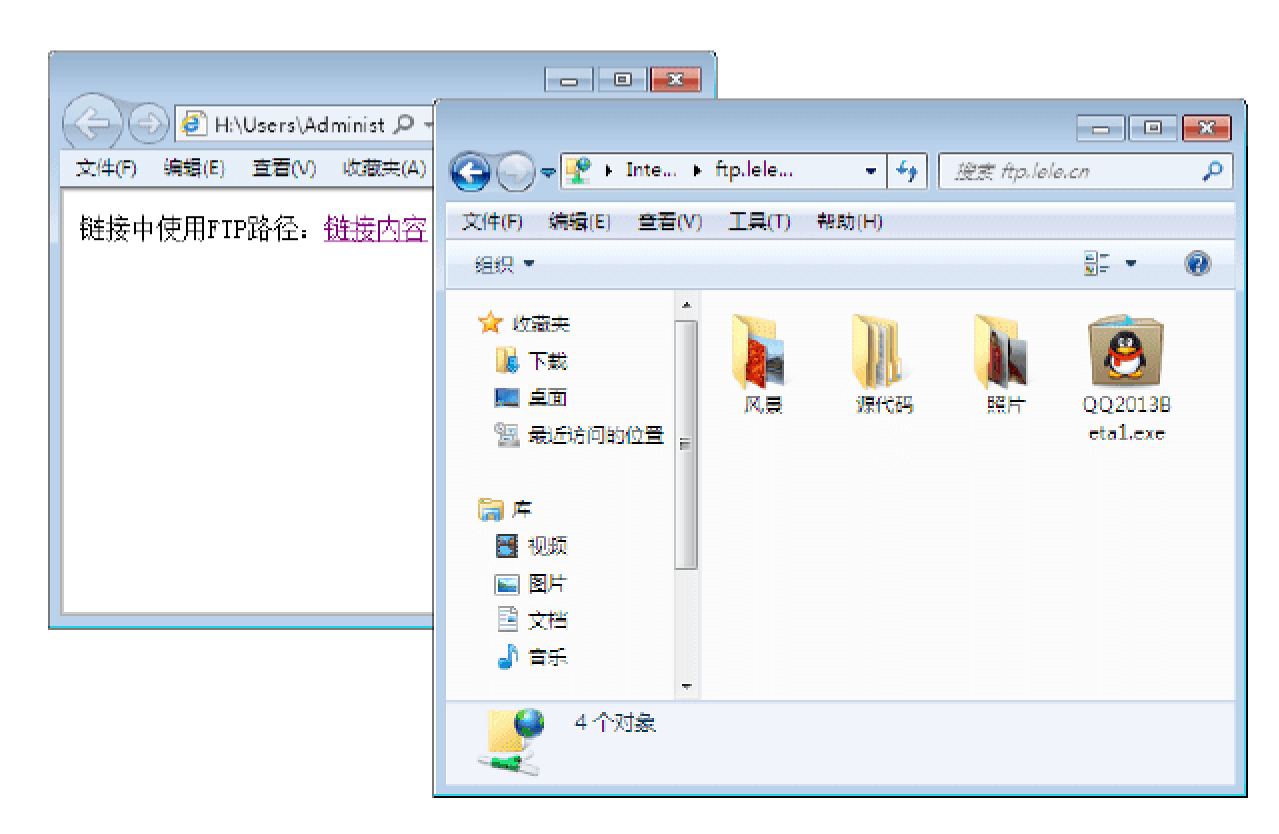


图 5-6 使用 FTP 路径的显示效果

5.1.5 电子邮件路径

电子邮件路径用来链接一个电子邮件的地址。下面是邮件路径可以使用的写法。

```
mailto:邮件地址
```

【例 5.4】（实例文件：ch05\5.4.html）

```
<html>
<head>
<title>电子邮件路径</title>
</head>
<body>
使用电子邮件路径：<a href="mailto:liule2012@163.com">链接</a>
</body>
</html>
```

其代码运行后，当单击含有链接的文本时，会弹出一个发送邮件的对话框，显示效果如图 5-7 所示。

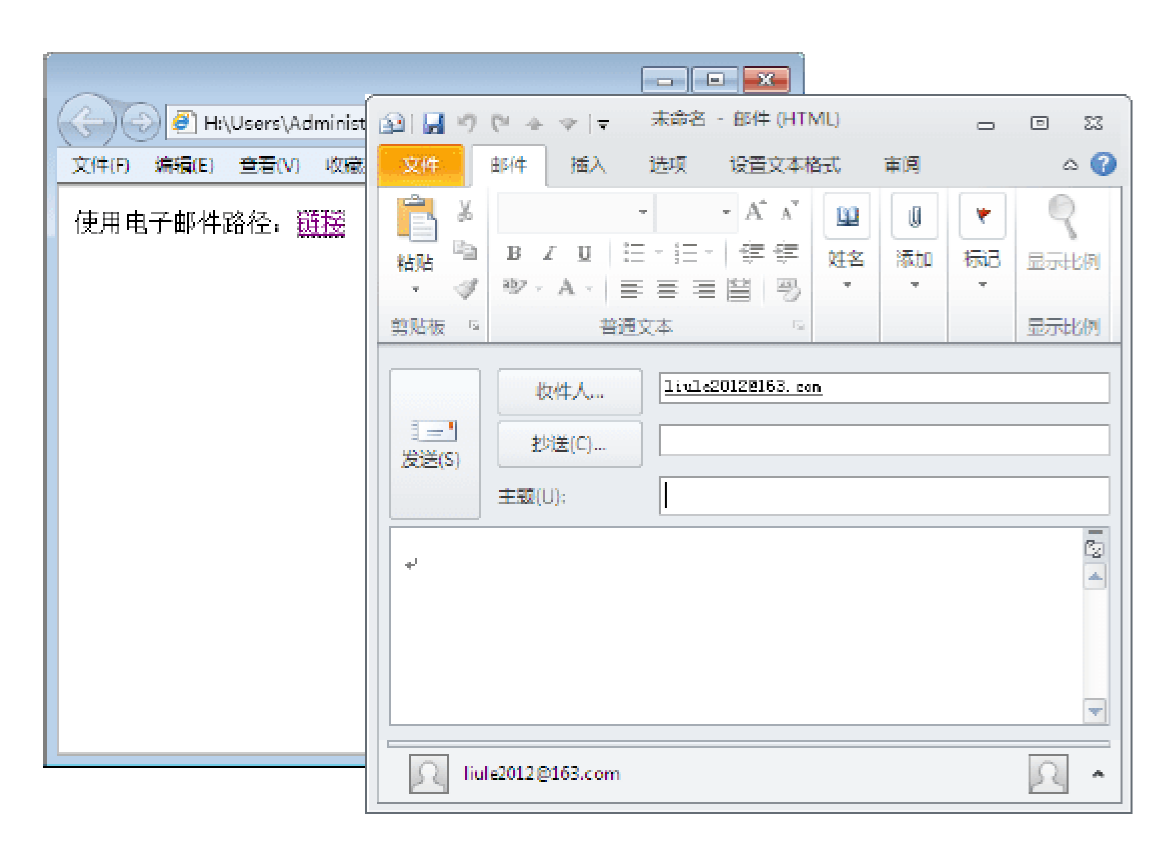


图 5-7 使用电子邮件路径的显示效果

5.2 链接元素

在 HTML 中，链接元素为<a>，<a>标签可定义锚。锚（anchor）有两种用法：

- 通过使用 href 属性，创建指向另外一个文档的链接（或超链接）。
- 通过使用 name 或 id 属性，创建一个文档内部的书签（也就是说，可以创建指向文档片段的链接）。

5.2.1 链接元素<a>

链接元素<a>用来定义一个超链接，前面使用链接时已经用到<a>元素，其实在<a>元素中，不但可以包含文本内容，也可以包含图片等其他内容。语法结构如下所示。

```
<a href="链接的路径">链接的文本</a>
```

在<a>元素中，一般要指定 href 属性，用来指向链接的目标。

【例 5.5】（实例文件：ch05\5.5.html）

```
<html>
<head>
<title>链接元素<a></title>
</head>
<body>
链接元素：<a href="http://www.baidu.com">链接内容</a>
</body>
</html>
```

其代码运行后，显示效果如图 5-8 所示。当单击页面中含有链接的文本时，显示效果如图 5-9 所示。

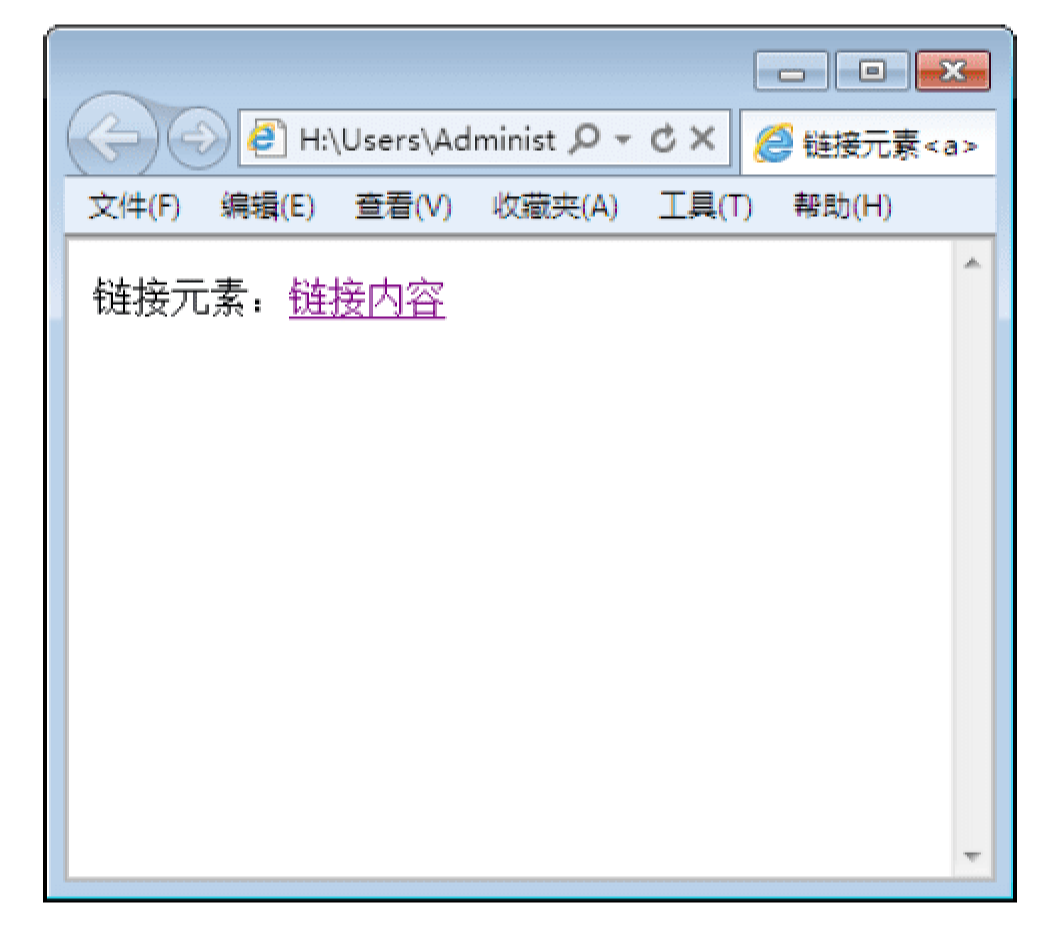


图 5-8 使用<a>元素的显示效果



图 5-9 单击链接文本后的显示效果



此时由于<a>元素中没有定义其他任何属性，所以链接的目标页面，将在<a>元素所在的页面中打开，而不会弹出新的页面。

5.2.2　指定路径属性 href

指定路径属性 href 用来指定链接元素<a>的目标地址。语法结构如下。

```
<a href="链接的路径">链接的文本</a>
```

【例 5.6】（实例文件：ch05\5.6.html）

```
<html>
<head>
<title>指定路径属性 href </title>
</head>
<body>
链接元素：<a href="http://www.114la.com">链接内容</a>
</body>
</html>
```

其代码运行后，显示效果如图 5-10 所示。当单击链接的文本时，显示效果如图 5-11 所示。

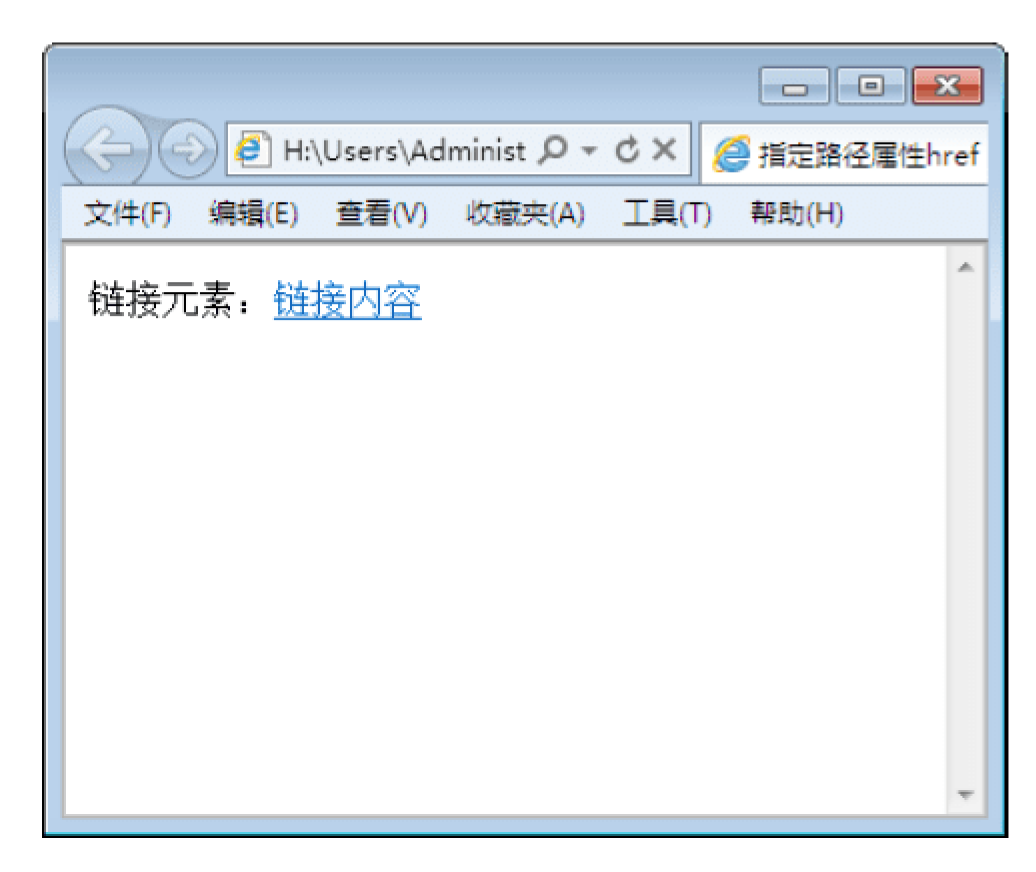


图 5-10　使用 href 属性的显示效果



图 5-11　单击链接文本后的显示效果

5.2.3　显示链接目标属性 target

显示链接目标属性 target 用来指定链接目标文件显示的窗口。语法结构如下。

```
<a href="链接的路径" target=”目标窗口或指定值”>链接的文本</a>
```

target 属性的取值有_self、_blank、_parent 和_top。

下面给出一个实例，其 target 取值为_self。

【例 5.7】（实例文件：ch05\5.7.html）

```
<html>
<head>
<title>显示链接目标属性 1</title>
</head>
<body>
链接元素：链接内容<a href="http://www.baidu.com/" target="_self">链接内容</a>
</body>
</html>
```

其代码运行后，显示效果如图 5-12 所示。当单击页面中含有链接的文本时，页面跳转，显示效果如图 5-13 所示。

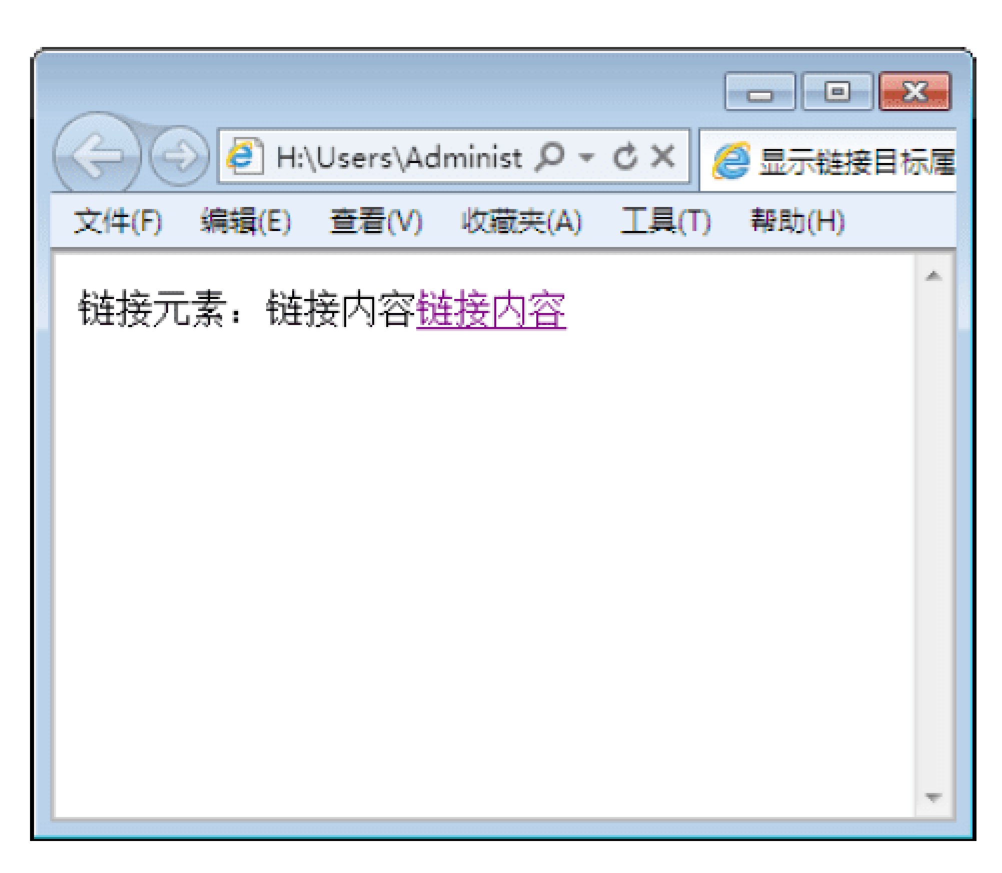


图 5-12 使用 target 属性值为_self 的显示效果



图 5-13 单击链接文本后的显示效果

当 target 属性取值为_blank 时，下面给出一个实例。

【例 5.8】（实例文件：ch05\5.8.html）

```
<html>
<head>
<title>显示链接目标属性 2</title>
</head>
<body>
链接元素：链接地址<a href="http://www.baidu.com/" target="_blank">链接地址</a>
</body>
</html>
```

其代码运行后，当单击含有链接的内容时，新的内容会在新的窗口显示，而不会覆盖原窗口的内容，其显示效果如图 5-14 所示。



图 5-14　使用 target 属性值为_blank 的显示效果



当 target 属性取值为_parent 和_top 时，其显示效果和使用_self 值相同，均为在当前窗口中显示目标页面。

5.2.4　激活顺序属性 tabindex

激活顺序属性 tabindex，用来指定当按下 Tab 键时，页面中链接的激活顺序。语法结构如下。

```
<a href="链接的路径" tabindex= "数字值">链接的文本</a>
```



tabindex 属性的取值是大于 0 的整数。下面是一个使用 tabindex 属性的实例。

【例 5.9】（实例文件：ch05\5.9.html）

```
<html>
<head>
<title>激活顺序属性</title>
</head>
<body>
链接路径 1：<a href="http://www.baidu.com" tabindex="2">链接内容</a><br />
链接路径 2：<a href="http://www.114la.com" tabindex="3">链接内容</a><br />
链接路径 3：<a href="http://www.sina.com" tabindex="1">链接内容</a>
</body>
</html>
```

其代码运行后，按下键盘的 Tab 键，可以依次激活页面中的链接，如图 5-15 所示。

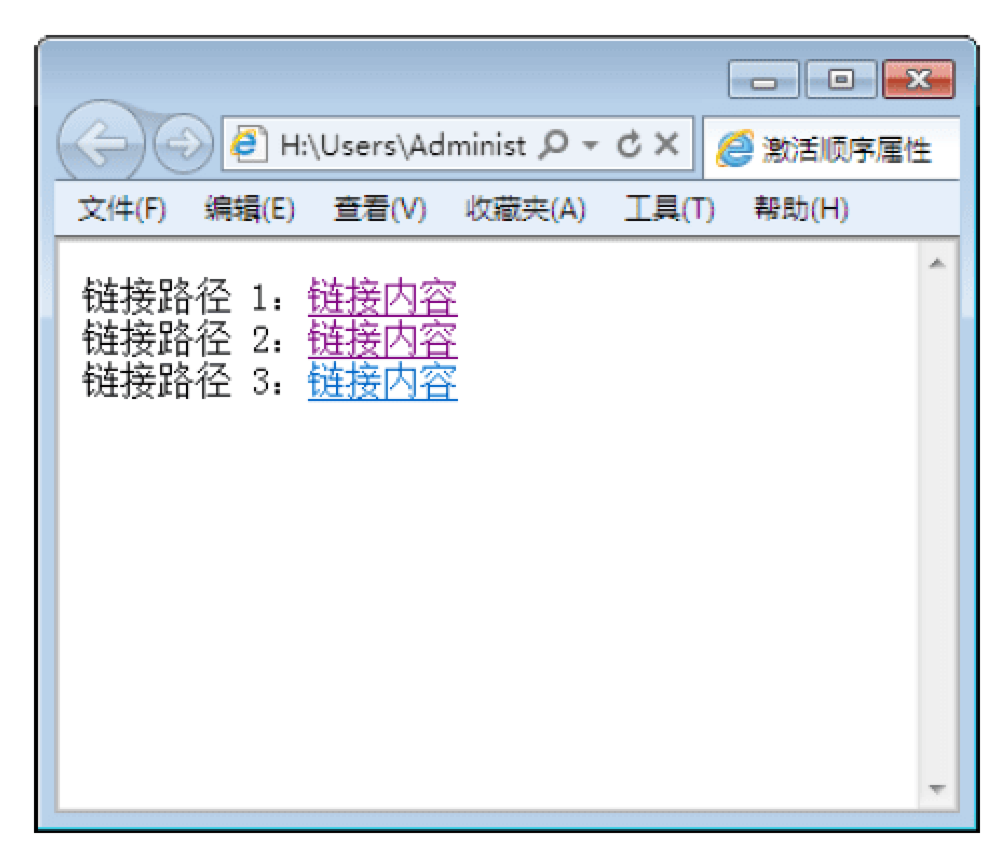


图 5-15　使用 tabindex 属性的显示效果

当按下 Tab 键时，页面中链接元素 2 被激活，如图 5-16 所示。此时，按下 Enter 键打开链接，其显示效果如图 5-17 所示。

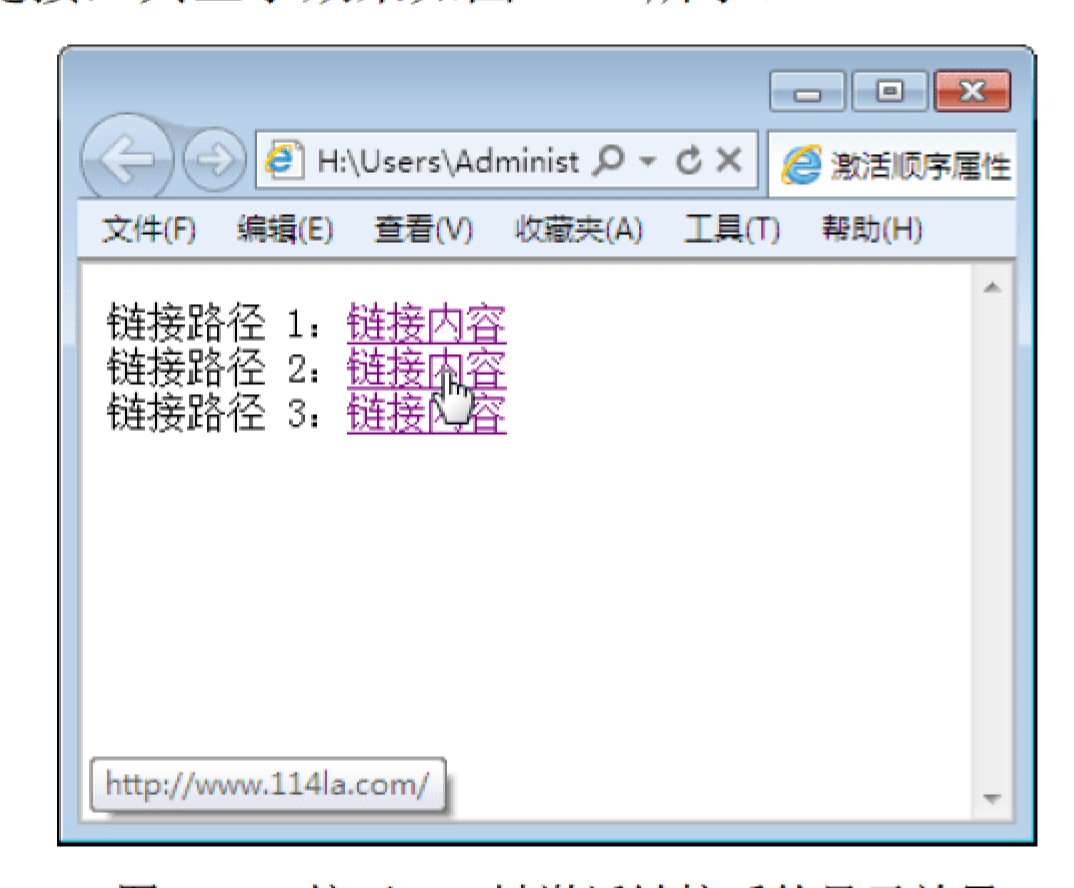


图 5-16　按下 Tab 键激活链接后的显示效果



图 5-17　按下 Enter 键打开链接后的显示效果

5.2.5　链接的热键属性 accesskey

链接的热键属性 accesskey 用来指定激活链接所使用的键。语法结构如下。

```
<a href="链接的路径" accesskey= "键盘上的键值">链接的文本</a>
```

由于用户使用的操作系统不同，要在键盘上按下相应的键（Window 中为 Alt 键），同时按下定义的热键，才能激活定义的链接。链接激活后，按下键盘的 Enter 键，可打开相应的链接。

下面是一个使用 accesskey 属性的实例。

【例 5.10】（实例文件：ch05\5.10.html）

```
<html>
<head>
<title>激活顺序属性</title>
```

```
</head>
<body>
链接元素：<a href="http://www.baidu.com" accesskey="z">链接内容</a>
</body>
</html>
```

其代码运行后，显示效果如图 5-18 所示。当同时按下键盘上的 Alt 和 Z 键时，链接文本被激活，其显示效果如图 5-19 所示。

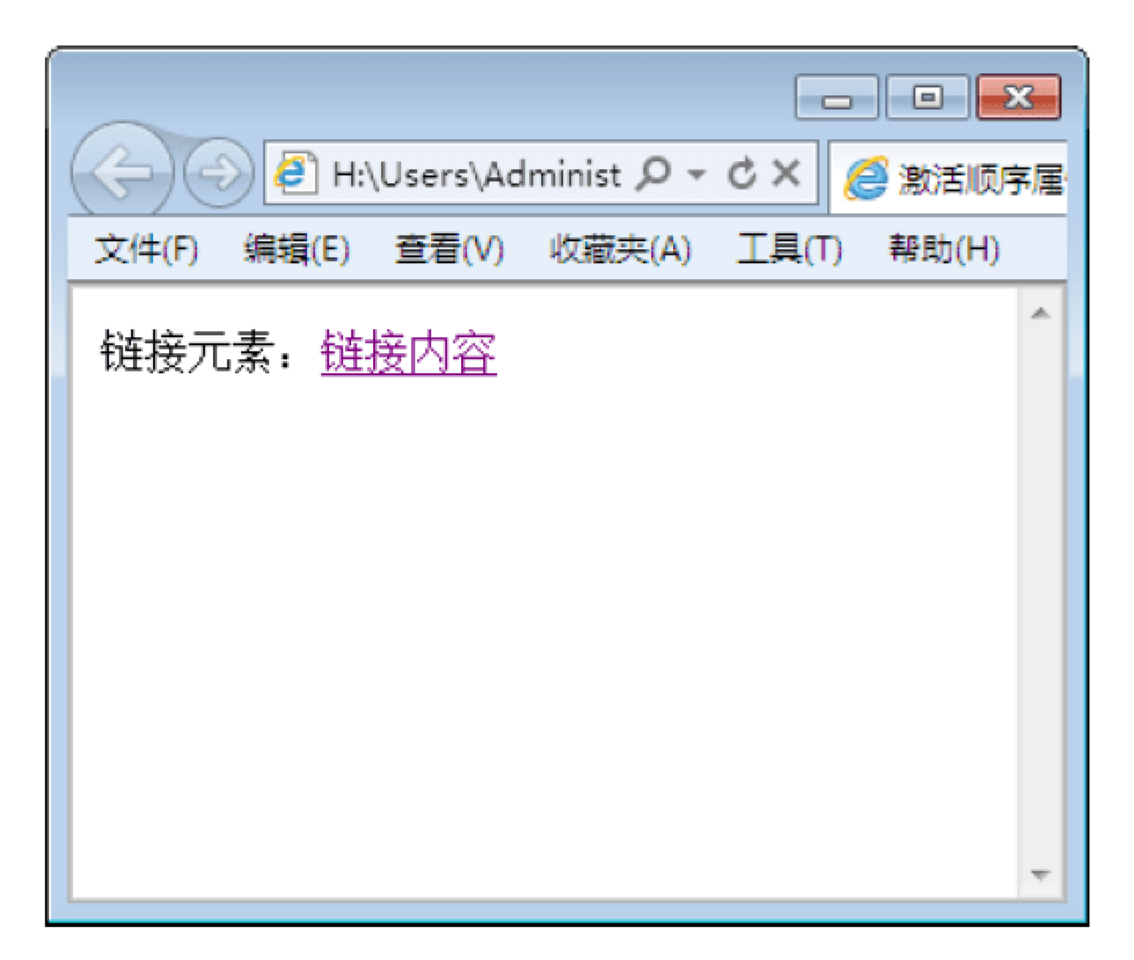


图 5-18　使用 accesskey 属性的显示效果

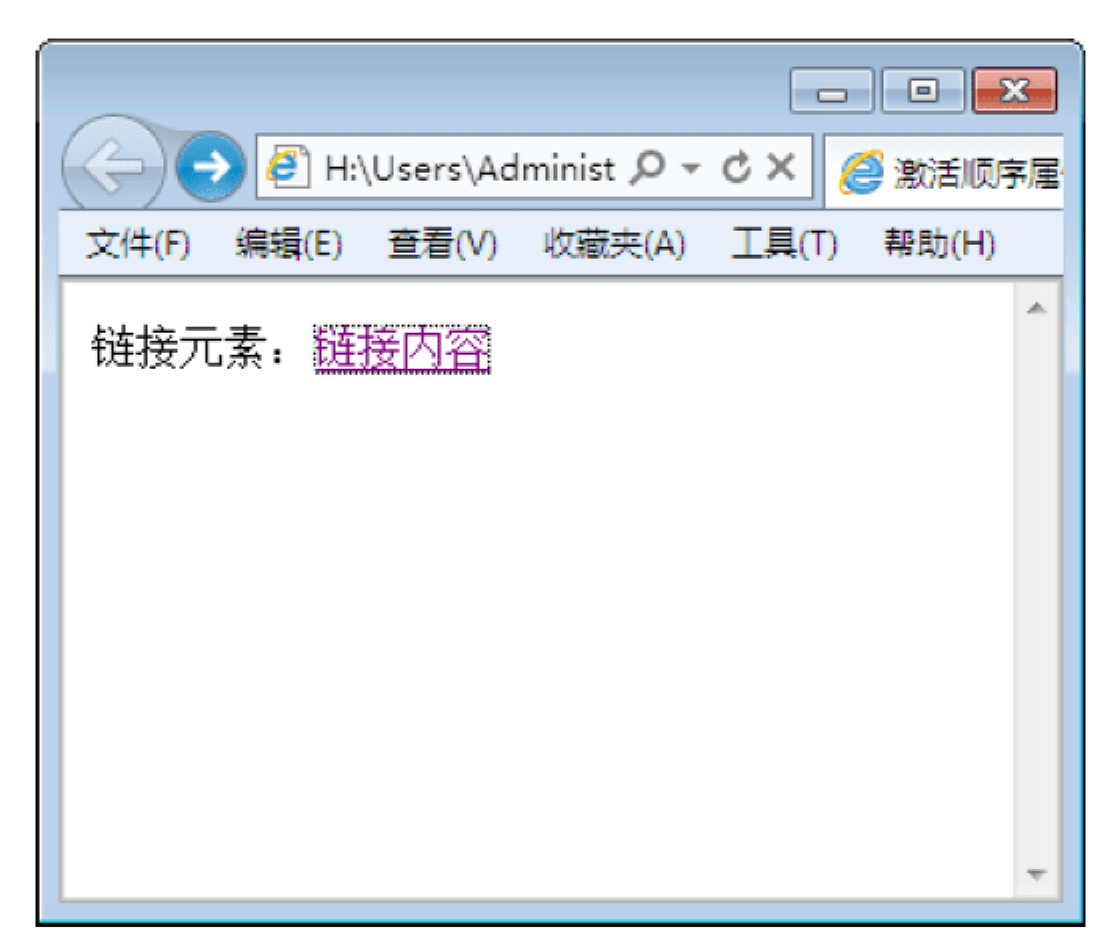


图 5-19　链接文本被激活后的显示效果

从图 5-19 可以看出，激活的链接文本，会在文本上产生一个虚线框，用来表明该链接已经被激活。当按下 Enter 键时，打开链接的页面，如图 5-20 所示。



图 5-20　按下 Enter 键打开链接后的显示效果

5.2.6　链接到同一页面的不同位置

在建立链接时，尤其是文字比较多的网页，需要在同一页面的不同位置进行链接，这时就

需要建立同一网页内的链接。语法结构如下。

```
<a href="#链接位置">链接文字</a>
```

【例 5.11】（实例文件：ch05\5.11.html）

```
<html>
<body>
<p>
<a href="#C4">查看 第 4 章。</a>
</p>
<h2>第 1 章</h2>
<p>本章讲解图片相关知识……</p>
<h2>第 2 章</h2>
<p>本章讲解文字相关知识……</p>
<h2>第 3 章</h2>
<p>本章讲解动画相关知识……</p>
<h2><a name="C4">第 4 章</a></h2>
<p>本章讲解图形相关知识……</p>
<h2>第 5 章</h2>
<p>本章讲解列表相关知识……</p>
<h2>第 6 章</h2>
<p>本章讲解列表相关知识……</p>
</body>
</html>
```

在 IE 9.0 中预览网页效果如图 5-21 所示。单击页面中的链接，即可将“第 4 章”的内容跳转到页面顶部，如图 5-22 所示。



图 5-21 链接页面

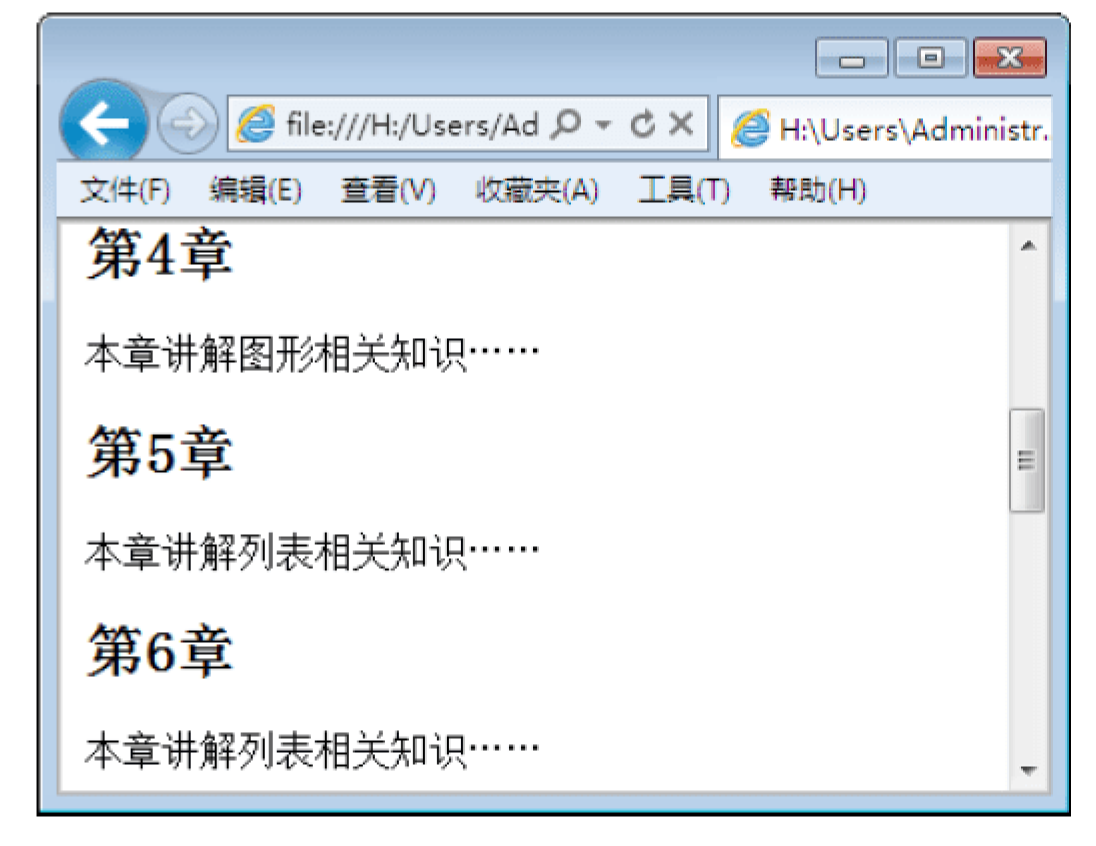


图 5-22 链接到“第 4 章”

5.3 图像链接

图像中的链接包括为图像元素制作的链接和在图像局部制作的链接。其中在图像的局部制作链接比较复杂，会用到<map>、<area>等元素及其相关属性。

5.3.1 创建链接区域元素<map>

<map>元素的主要作用是标记链接区域。<map>元素本身并不能指定链接区域的大小和链接目标。页面中的图像元素可以使用<map>元素标记的区域。语法结构如下。

```
<map>其他元素</map>
```



<map>元素一般要结合<area>元素一起使用。

5.3.2 链接区域的名称属性 name

链接区域的名称属性 name 用来定义链接区域的名称，以方便图像元素的调用。语法结构如下。

```
<map name="名称">其他元素</map>
```



name 属性的取值必须是唯一的。

5.3.3 定义鼠标敏感区元素 area

定义鼠标敏感区元素<area>用来定义链接区域的大小和坐标，同时可以指定每个敏感区域的链接目标。语法结构如下。

```
<map name="名称">
<area  属性="属性值" />
</map>
```

下面给出一个使用<area>元素的实例。

【例 5.12】（实例文件：ch05\5.12.html）

```
<html>
<head>
<title>鼠标敏感区</title>
</head>
<body>
<img src="小花.jpg" alt="郁金香" border="0" usemap="#Map" />
<map name="Map" id="Map">
<area shape="rect" coords="27,20,132,122" href="http://www.baidu.com" />
```

```
<area shape="rect" coords="226,61,322,200" href="http://www.hao123.com" />
<area shape="circle" coords="114,235,52" href="http://www.sina.com" />
</map>
</body>
</html>
```

该实例中，在图片中定义了三个链接区域，分别链接到百度、hao123 和新浪的站点首页。在图片的局部制作链接后，对图片的显示效果并没有影响。其代码运行后，显示效果如图 5-23 所示。

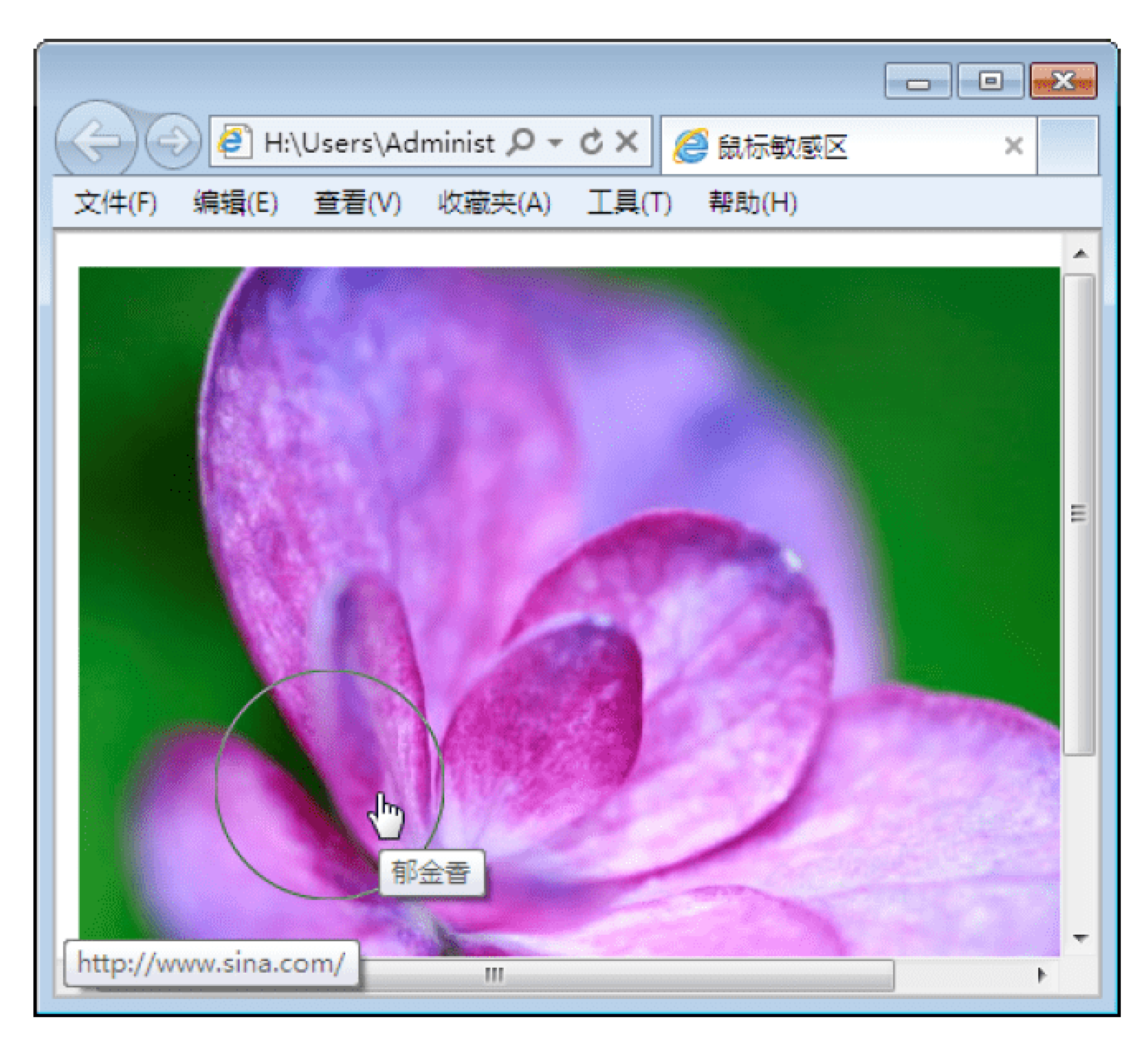


图 5-23　局部定义链接后的图像元素

5.3.4　链接的路径属性 href

指定路径属性 href 用来指定链接的地址。不指定路径属性 nohref 用来不指定链接的地址。其中 href 属性和<a>元素中的 href 属性完全相同。使用指定路径属性 href 的语法如下。

```
<map name="名称">
<area href="指定的路径" />
</map>
```

不指定路径属性 nohref 的语法如下。

```
<map name="名称">
<area nohref="nohref" />
</map>
```

下面给出一个使用< area nohref >元素的实例。

【例 5.13】（实例文件：ch05\5.13.html）

```
<html>
<head>
<title>< area nohref >元素</title>
</head>
<body>
<img src="小花.jpg" alt="郁金香" border="0" usemap="#Map" />
<map name="Map" id="Map">
<area shape="rect" coords="27,20,132,122" nohref="http://www.baidu.com" />
<area shape="rect" coords="226,61,322,200"nohref="http://www.hao123.com" />
<area shape="circle" coords="114,235,52" nohref="http://www.sina.com" />
</map>
</body>
</html>
```

使用了 nohref 属性的区域，当鼠标悬停的时候，将不再显示手的形状。其代码运行后，显示效果如图 5-24 所示。

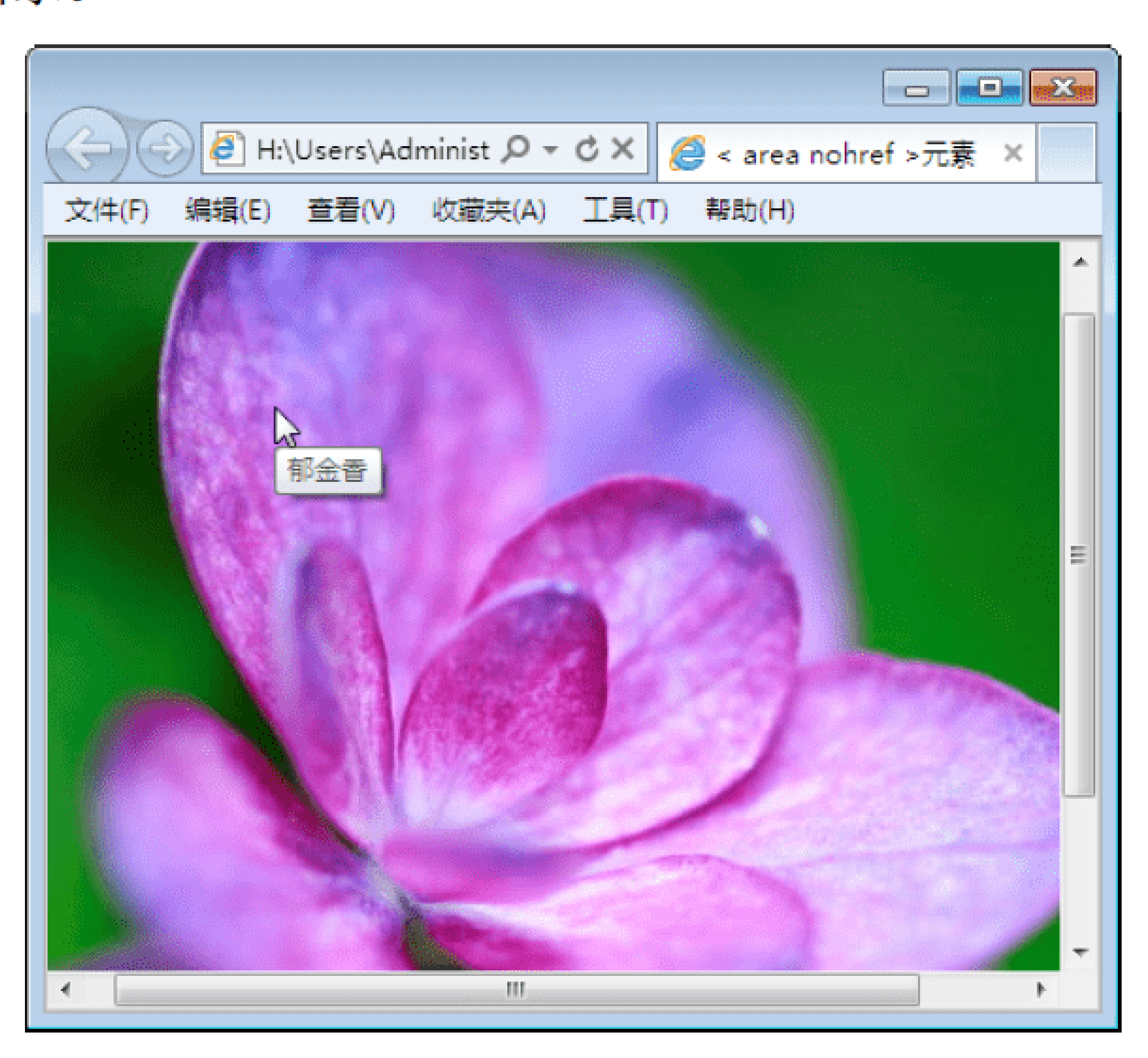


图 5-24　使用< area nohref >元素链接后的图像元素

5.3.5　链接的文本说明属性 alt

链接的文本说明属性 alt 用来使用附加的文本对链接进行说明。和<img>元素里的 alt 属性类似，当鼠标悬停在某个链接区域时，将会显示出 alt 属性中附加的文本。语法结构如下。

```
<map name="名称">
<area alt="附加的文本" />
</map>
```

下面给出一个在<area>元素使用 alt 属性的实例。

【例 5.14】（实例文件：ch05\5.14.html）

```
<html>
<head>
<title>链接的文本说明</title>
</head>
<body>
<img src="小花.jpg" alt="郁金香" border="0" usemap="#Map" />
<map name="Map" id="Map">
<area shape="rect" coords="30,21,132,112" href="http://www.baidu.com" alt="左侧链接百度" />
<area shape="rect" coords="221,62,312,210" href="http://www.hao123.com" alt="右侧链接 hao123" />
<area shape="circle" coords="104,245,54" href="http://www.sina.com" alt="底部链接新浪" />
</map>
</body>
</html>
```

其代码运行后，当鼠标悬停在相应区域时，会出现相应的链接，显示效果如图 5-25 所示。

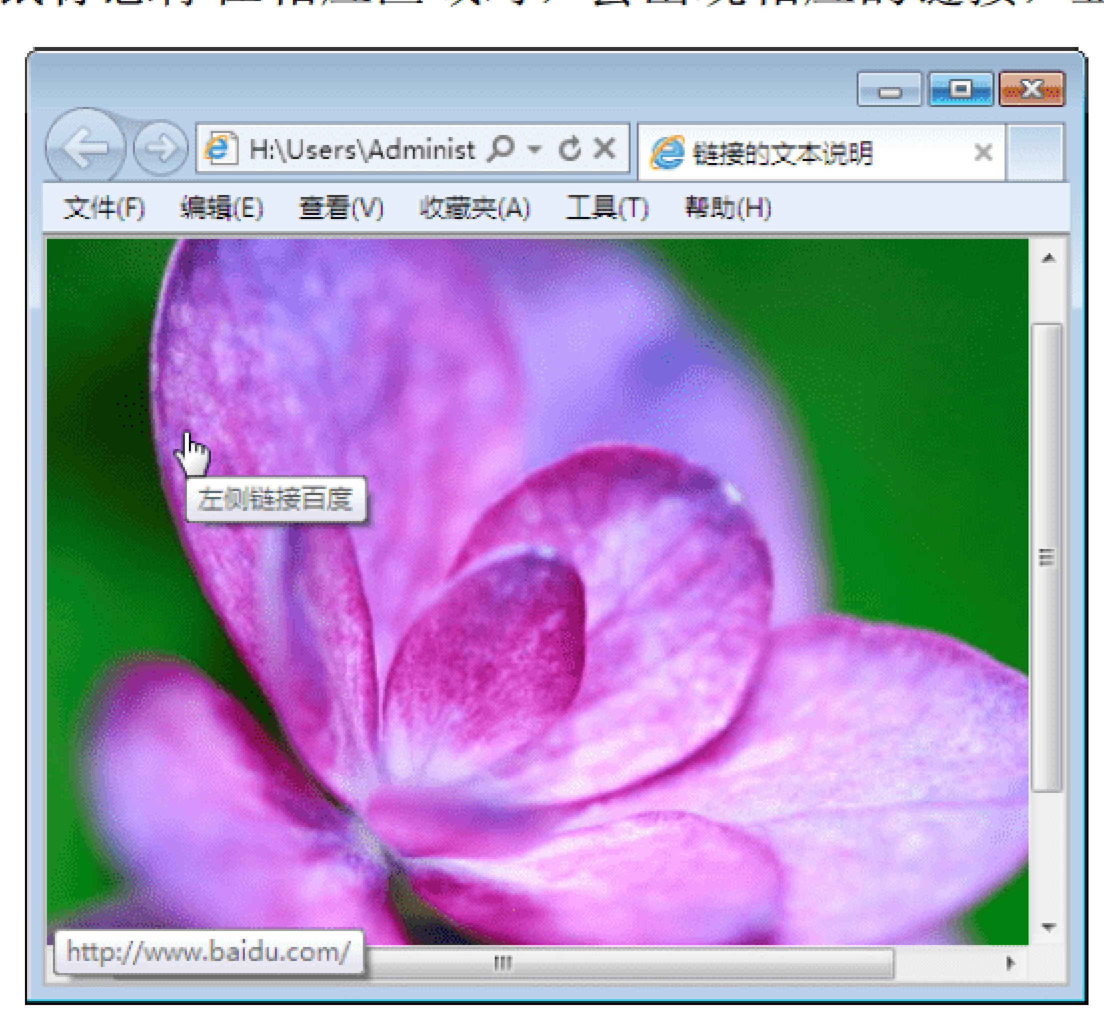


图 5-25　在<area>元素中使用 alt 属性的图像元素

5.4　专家解惑

1．在创建超链接时，使用绝对 URL 还是相对 URL？

答：在创建超链接时，如果要链接的是另外一个网站中的资源，需要使用完整的绝对 URL；如果在网页中创建内部链接，一般使用相对当前文档或站点根文件夹的相对 URL。

2．链接增多后，网站如何设置目录结构以方便维护？

答：当一个网站的网页数量增加到一定程度以后，网站的管理与维护将变得非常繁琐，因

此掌握一些网站管理与维护技术是非常实用的，可以节省很多时间。建立适合的网站文件存储结构，可以便于网站的管理与维护。通常使用的 3 种网站文件组织结构方案及文件管理遵循的原则如下。

（1）按照文件的类型进行分类管理。将不同类型的文件放在不同的文件夹中，这种存储方法适合于中小型的网站，这种方法通过文件的类型对文件进行管理。

（2）按照主题对文件进行分类。网站的页面按照不同的主题进行分类储存。将同一主题的所有文件存放在一个文件夹中，然后再进一步细分文件的类型。这种方案适用于页面与文件数量众多、信息量大的静态网站。

（3）对文件类型进行进一步细分存储管理。这种方案是第一种存储方案的深化，将页面进一步细分后进行分类存储管理。这种方案适用于文件类型复杂、包含各种文件的多媒体动态网站。

第 6 章

网页表格的设计

HTML 中的表格不但可以清晰地显示数据，而且可以用于页面布局。HTML 中的表格类似于 Word 软件中的表格，尤其是使用网页制作工具时操作很相似。

6.1 表格基本结构

HTML 制作表格的原理是使用相关标记，如表格对象 table 标记、行对象 tr 标记和单元格对象 td 标记。

6.1.1 表格概述

使用表格显示数据更直观、更清晰。在 HTML 文档中表格主要用于显示数据，虽然可以使用表格布局，但是这里不建议使用，它有很多弊端。表格一般由行、列和单元格组成，如图 6-1 所示。

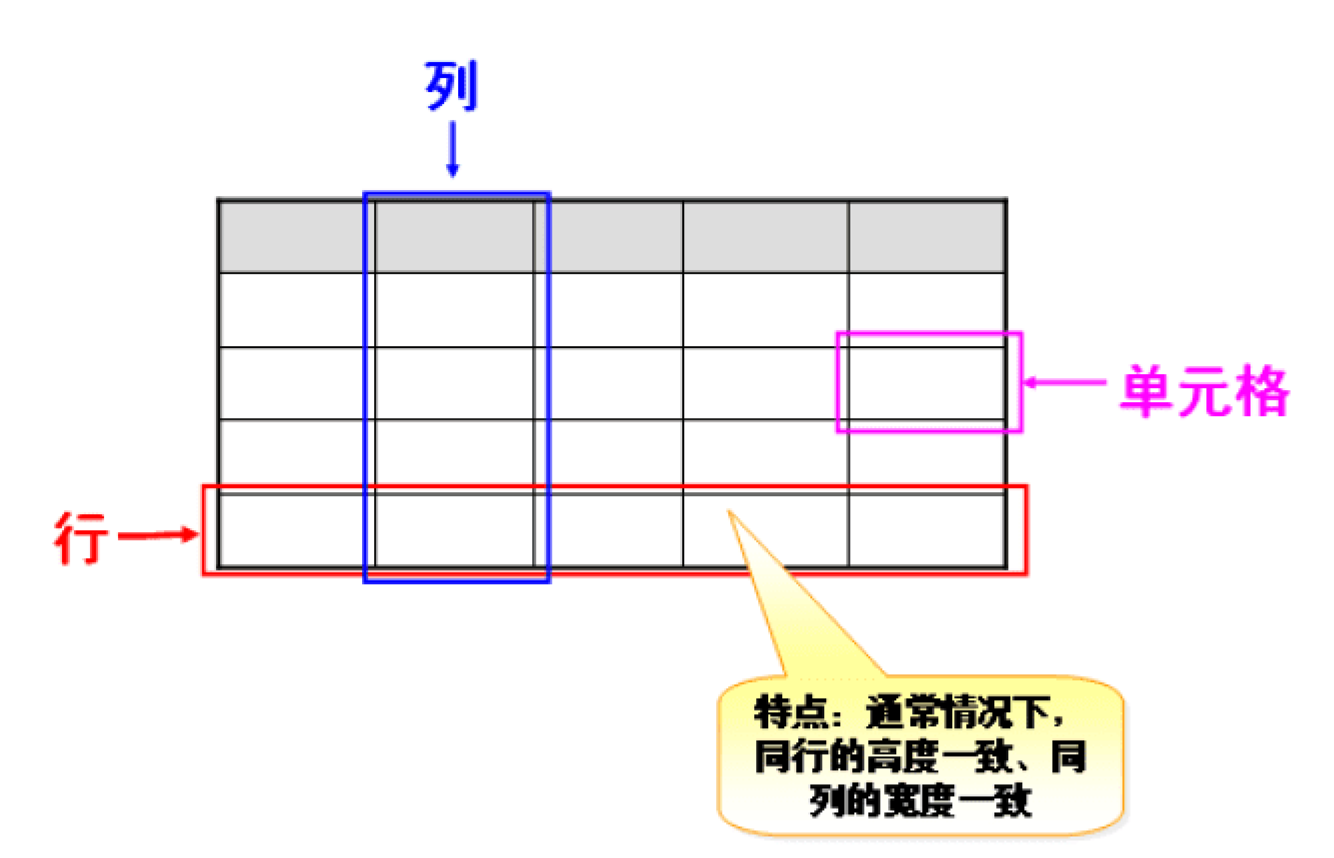


图 6-1 表格的组成

6.1.2 用于创建表格的标记

在 HTML 中用于创建表格的标记如下。

- <table>标记：用于标识一个表格对象的开始，</table>标记用于标识一个表格对象的结

束。一个表格中，只允许出现一对<table>标记。

- <tr>标记：用于标识表格一行的开始，</tr>标记用于标识表格一行的结束。表格内有多少对<tr></tr>标记，就表示表格中有多少行。
- <td>标记：用于标识表格某行中的一个单元格开始，</td>标记用于标识表格某行中的一个单元格结束。<td></td>标记书写在<tr></tr>标记内，一对<tr></tr>标记内有多少对<td></td>标记，就表示该行有多少个单元格。

最基本的表格必须包含一对<table></table>标记、一对或几对<tr></tr>标记以及一对或几对<td></td>标记。一对<table></table>标记定义一个表格，一对<tr></tr>标记定义一行，一对<td></td>标记定义一个单元格。

下面的实例定义了一个 4 行 3 列的表格。

【例 6.1】（实例文件：ch06\6.1.html）

```
<html>
<head>
<title>表格基本结构</title>
</head>
<body>
<table border="1">
  <tr>
    <td>A1</td>
    <td>B1</td>
    <td>C1</td>
  </tr>
  <tr>
    <td>A2</td>
    <td>B2</td>
    <td>C2</td>
  </tr>
  <tr>
    <td>A3</td>
    <td>B3</td>
    <td>C3</td>
  </tr>
  <tr>
    <td>A4</td>
    <td>B4</td>
    <td>C4</td>
  </tr>
</table>
</body>
</html>
```

在 IE 9.0 中预览网页效果如图 6-2 所示。

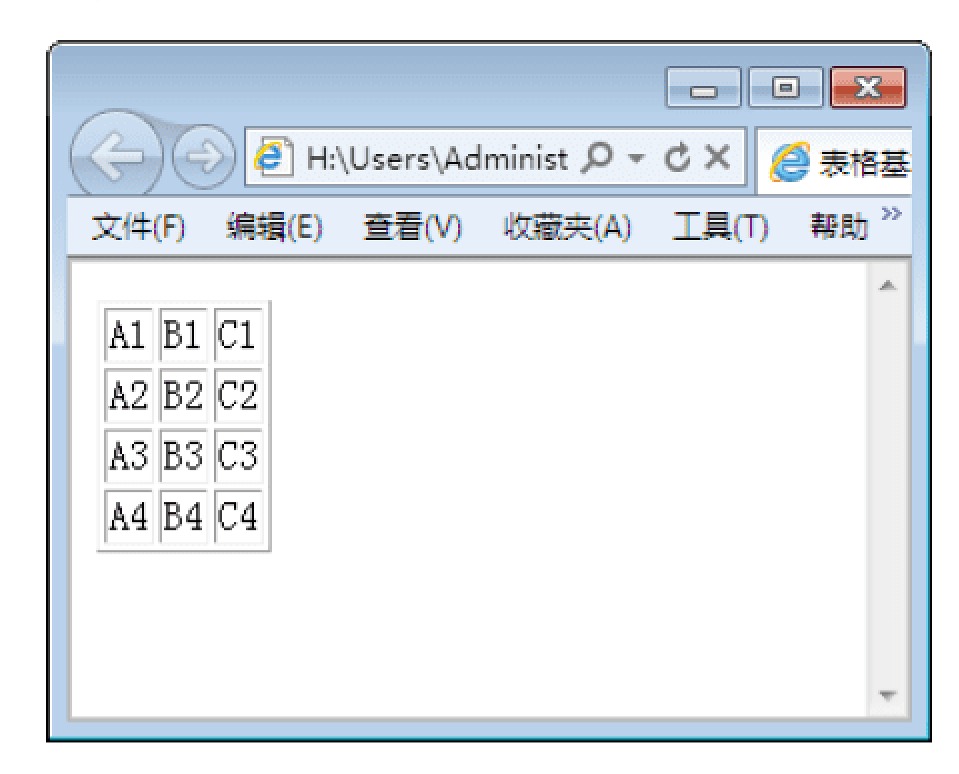


图 6-2 表格基本结构



从图 6-2 中，读者会发现，表格没有边框，行高及列宽也无法控制。

6.2 创建表格

表格可以分为普通表格和带有标题的表格，在 HTML 中，可以创建这两种表格。

6.2.1 创建普通表格

下面的实例创建了一列、一行三列和两行三列 3 个表格。

【例 6.2】（实例文件：ch06\6.2.html）

```
<html>
<body>
<h4>一列：</h4>
<table border="1">
<tr>
  <td>100</td>
</tr>
</table>
<h4>一行三列：</h4>
<table border="1">
<tr>
  <td>100</td>
  <td>200</td>
  <td>300</td>
</tr>
</table>
<h4>两行三列：</h4>
<table border="1">
```

```
<tr>
  <td>100</td>
  <td>200</td>
  <td>300</td>
</tr>
<tr>
  <td>400</td>
  <td>500</td>
  <td>600</td>
</tr>
</table>
</body>
</html>
```

在 IE 9.0 中预览网页效果如图 6-3 所示。

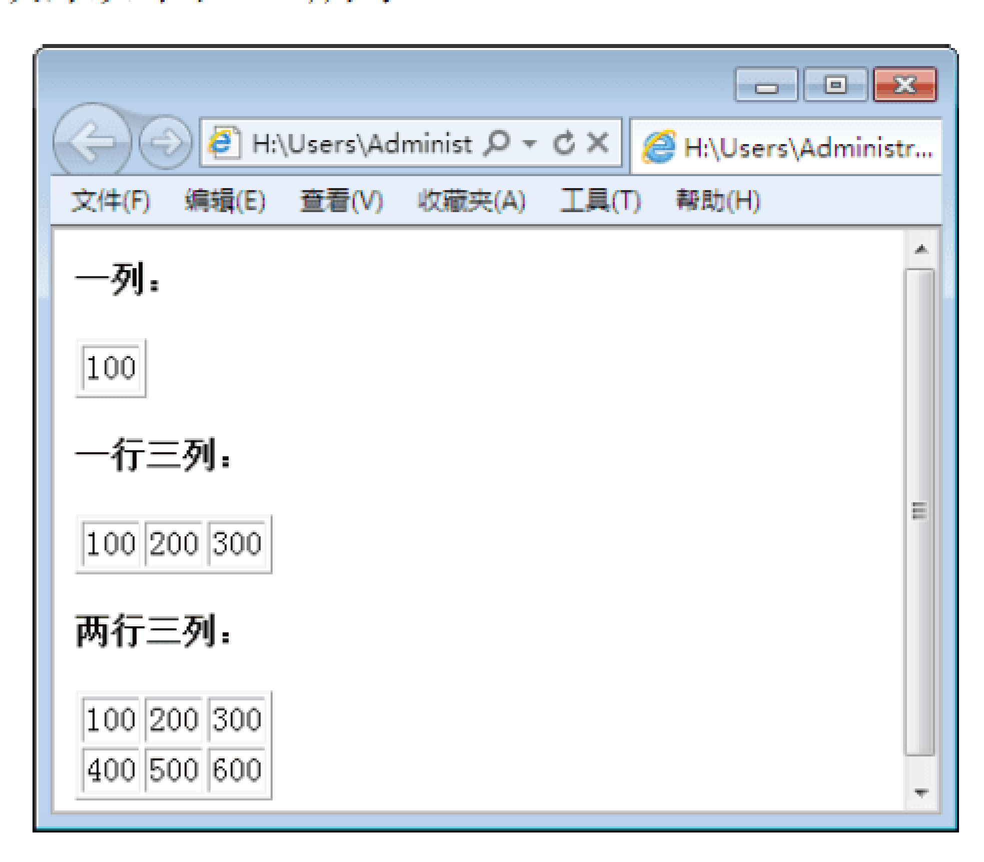


图 6-3 程序运行结果

6.2.2 创建带有标题的表格

有时，为了方便表述表格，还需要在表格的上面加上标题。下面的实例创建了一个带有标题的表格。

【例 6.3】（实例文件：ch06\6.3.html）

```
<html>
<body>
<h4>带有标题的表格</h4>
<table border="3">
<caption>数据统计表</caption>
<tr>
  <td>100</td>
  <td>200</td>
  <td>300</td>
```

```
</tr>
<tr>
  <td>400</td>
  <td>500</td>
  <td>600</td>
</tr>
</table>
</body>
</html>
```

在 IE 9.0 中预览网页效果如图 6-4 所示。

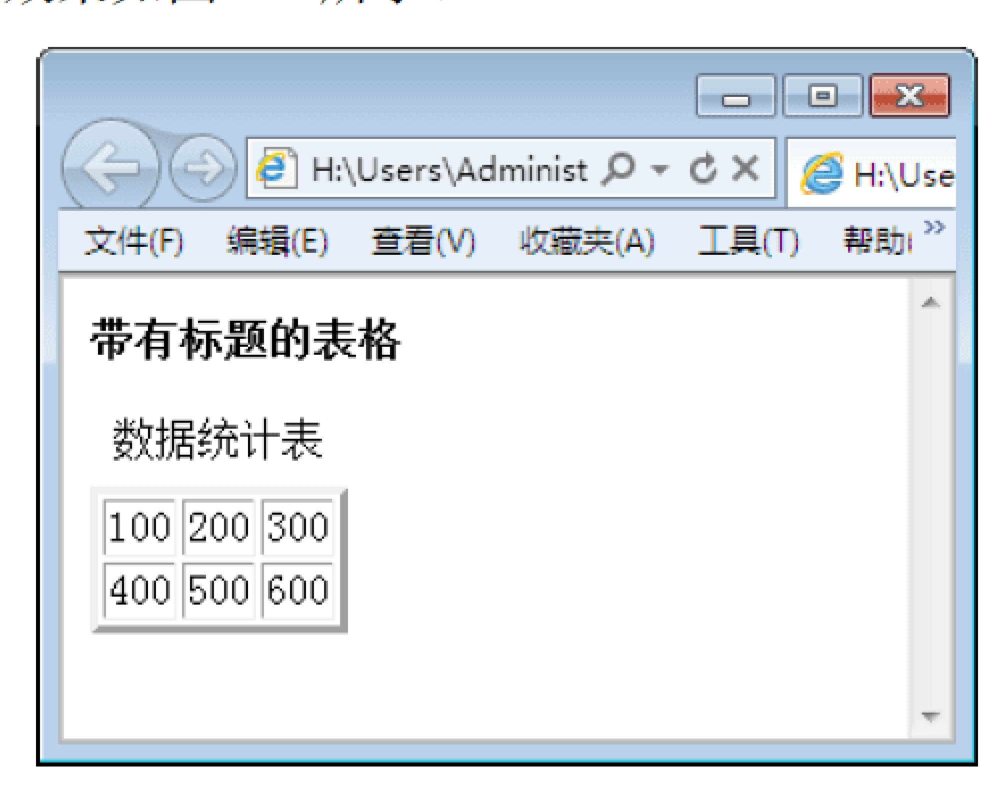


图 6-4　带有标题的表格

6.2.3　创建完整的表格

表格中最常用的 3 个标记是<table>、<tr>和<td>，使用它们可以构建出最简单的表格。为了让表格结构更清楚，并配合后面学习的 CSS 样式，更方便地制作各种样式的表格，表格中还会出现表头、主体、脚注等，如下面的实例所示。

【例 6.4】（实例文件：ch06\6.4.html）

```
<html>
<head>
<title>完整表格标记</title>
<style>
tfoot{
      background-color:#FF3;
      }
</style>
</head>
<body>
<table border="1">
  <caption>学生成绩单</caption>
```

```
    <thead>
      <tr>
        <th>姓名</th><th>性别</th><th>成绩</th>
      </tr>
    </thead>
     <tfoot>
      <tr>
        <td>平均分</td><td colspan="2">540</td>
      </tr>
    </tfoot>
    <tbody>
      <tr>
        <td>张三</td><td>男</td><td>560</td>
      </tr>
      <tr>
        <td>李四</td><td>男</td><td>520</td>
      </tr>
    </tbody>
  </table>
  </body>
  </html>
```

从上面的代码中可以发现，使用 caption 定义了表格标题，<thead>、<tbody>和<tfoot>标记对表格进行了分组。在<thead>部分使用<th>标记代替<td>标记定义单元格，<th>标记定义的单元格默认加粗。网页预览效果如图 6-5 所示。

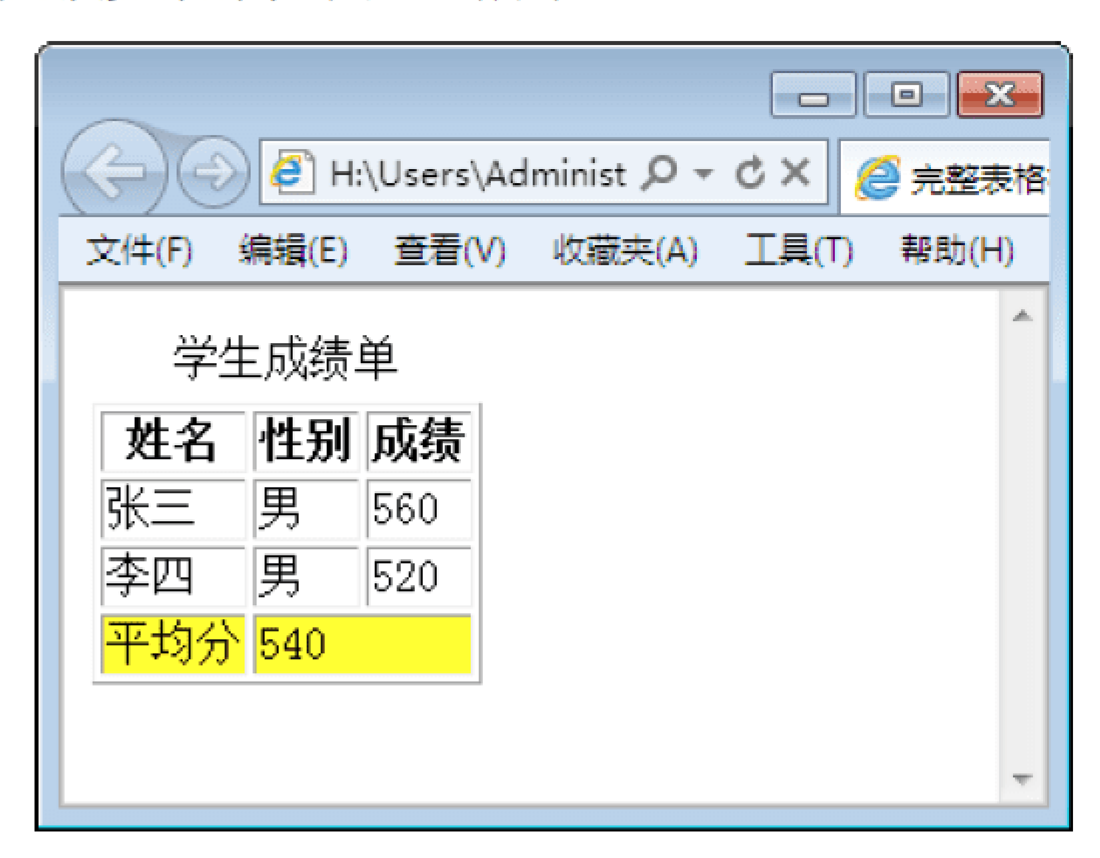


图 6-5　完整的表格结构



<caption> 标签必须紧随<table> 标签之后。

6.3 设置表格属性

在了解了表格的基本结构后，下面来介绍表格的基本操作，主要包括创建表格、设置表格的边框类型、设置表格的表头、合并单元等。

6.3.1 定义表格的边框类型

使用表格的 border 属性可以定义表格的边框类型，如常见的边框加租的表格。下面的实例创建了不同边框类型的表格。

【例 6.5】（实例文件：ch06\6.5.html）

```
<html>
<body>
<h4>普通边框</h4>
<table border="1">
<tr>
  <td>First</td>
  <td>Row</td>
</tr>
<tr>
  <td>Second</td>
  <td>Row</td>
</tr>
</table>
<h4>加粗边框</h4>
<table border="8">
<tr>
  <td>First</td>
  <td>Row</td>
</tr>
<tr>
  <td>Second</td>
  <td>Row</td>
</tr>
</table>
</body>
</html>
```

在 IE 9.0 中预览网页效果如图 6-6 所示。

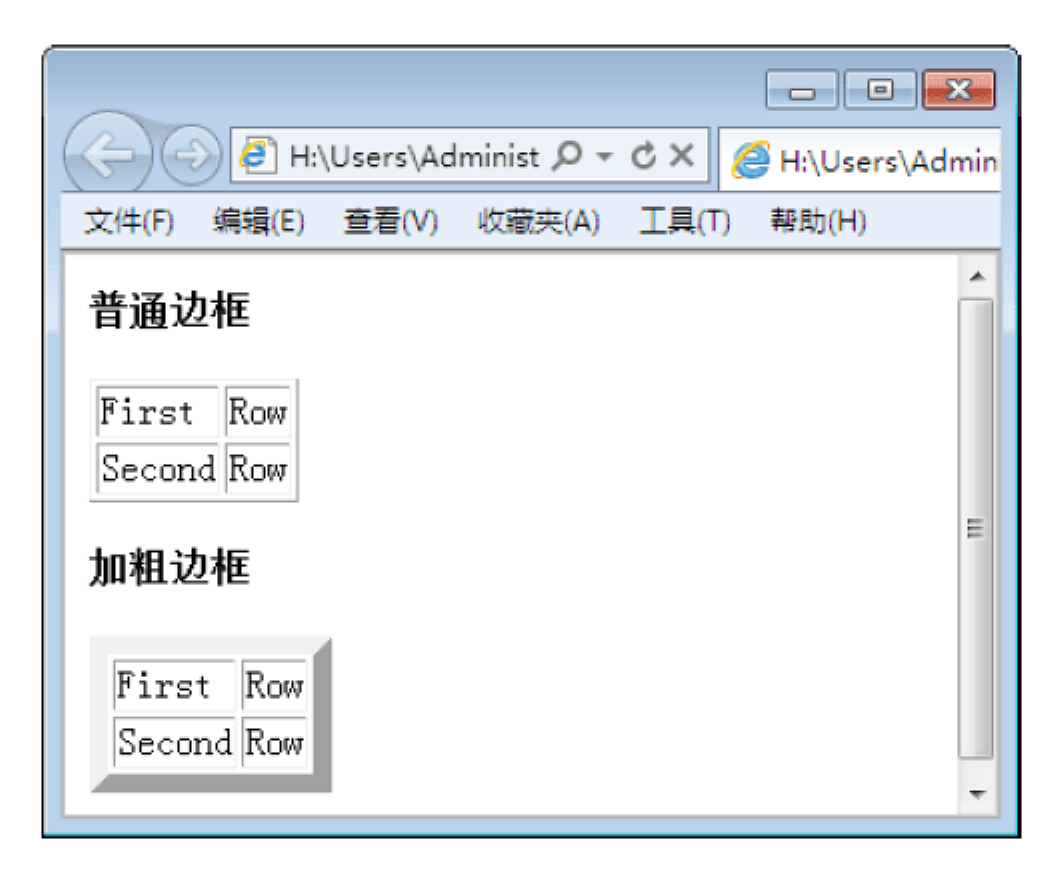


图 6-6　不同边框类型的表格

6.3.2　定义表格的表头

表格中也存在表头，常见的表头分为垂直与水平两种。下面的实例分别创建了带有垂直和水平表头的表格。

【例 6.6】（实例文件：ch06\6.6.html）

```
<html>
<body>
<h4>水平的表头</h4>
<table border="1">
<tr>
  <th>姓名</th>
  <th>性别</th>
  <th>电话</th>
</tr>
<tr>
  <td>张三</td>
  <td>男</td>
  <td>123456</td>
</tr>
</table>
<h4>垂直的表头</h4>
<table border="1">
<tr>
  <th>姓名</th>
  <td>小丽</td>
</tr>
<tr>
  <th>性别</th>
  <td>女</td>
```

```
</tr>
<tr>
  <th>电话</th>
  <td>123456</td>
</tr>
</table>
</body>
</html>
```

在 IE 9.0 中预览网页效果如图 6-7 所示。

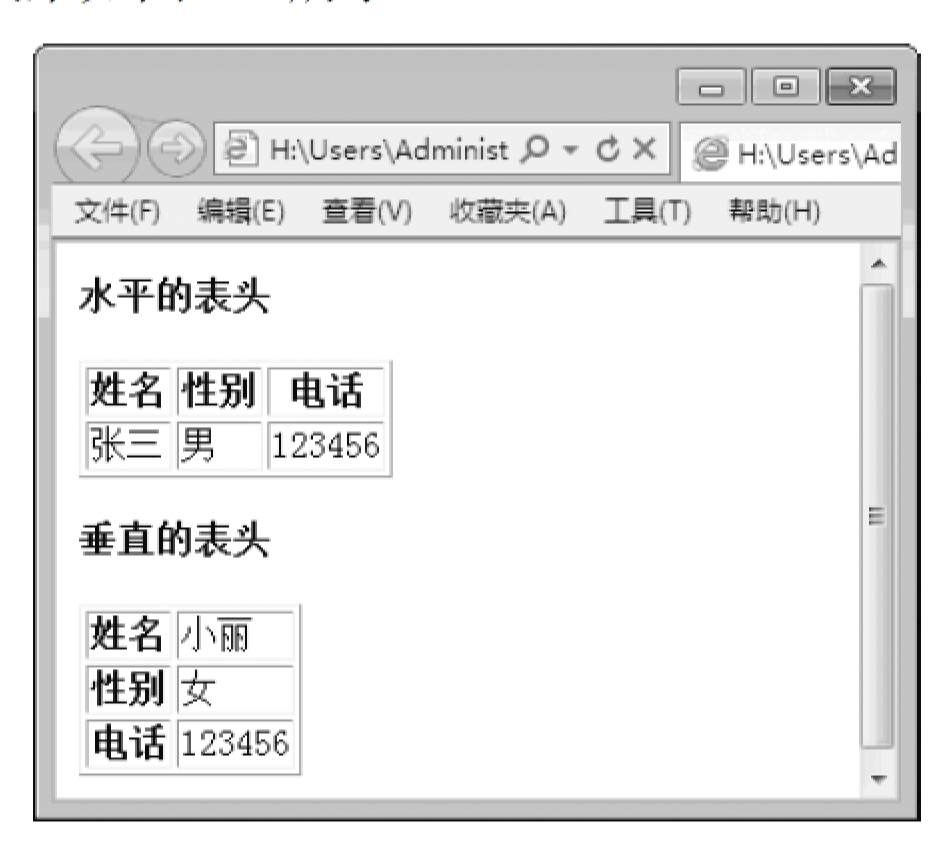


图 6-7　带有水平垂直表头的表格

6.3.3　设置表格背景

当创建好表格后，为了美观，还可以设置表格的背景。

1. 定义表格背景颜色

为表格添加背景颜色是美化表格的一种方式。下面的实例为表格添加背景颜色。

【例 6.7】（实例文件：ch06\6.7.html）

```
<html>
<body>
<h4>背景颜色：</h4>
<table border="1"
bgcolor="green">
<tr>
  <td>100</td>
  <td>200</td>
</tr>
<tr>
  <td>300</td>
  <td>400</td>
```

```
</tr>
</table>
</body>
</html>
```

在 IE 9.0 中预览网页效果如图 6-8 所示。

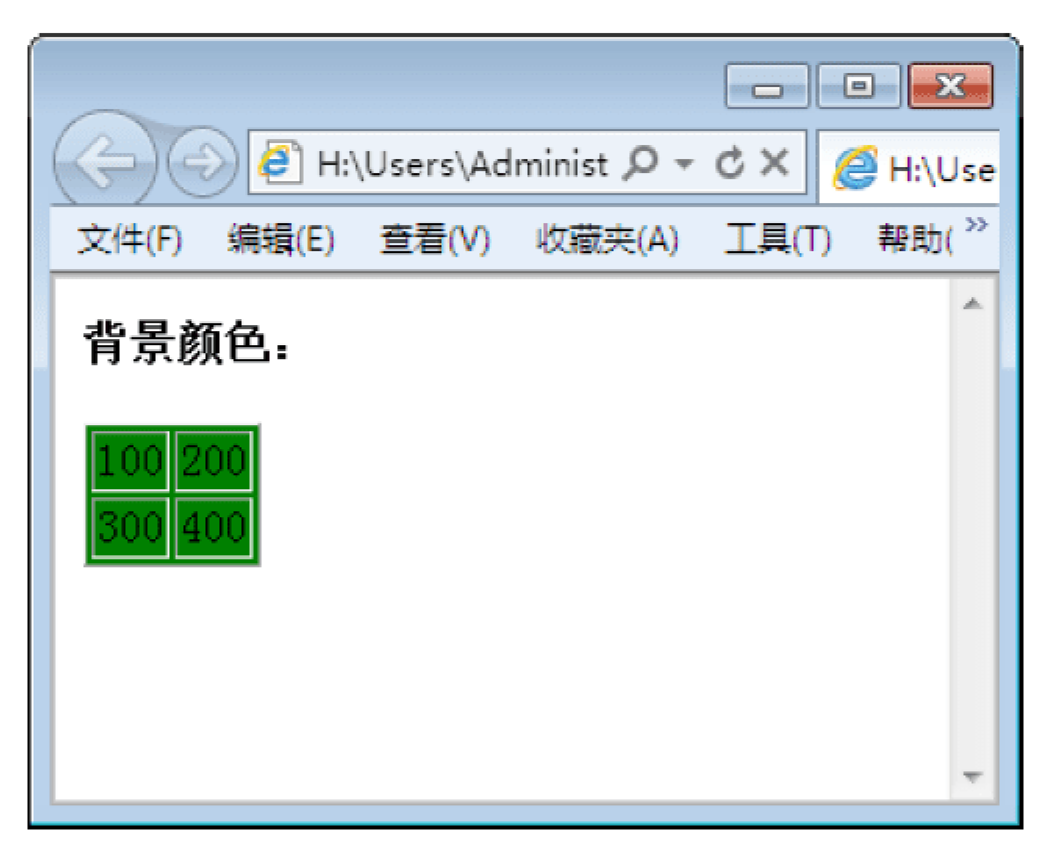


图 6-8 为表格添加背景颜色

2. 定义表格背景图片

除了可以为表格添加背景颜色外，还可以将图片设置为表格的背景。下面的实例为表格添加背景图片。

【例 6.8】（实例文件：ch06\6.8.html）

```
<html>
<body>
<h4>背景图片：</h4>
<table border="1"
background="images/1.gif">
<tr>
   <td>100</td>
   <td>200</td>
</tr>
<tr>
   <td>300</td>
   <td>400</td>
</tr>
</table>
</body>
</html>
```

在 IE 9.0 中预览网页效果如图 6-9 所示。

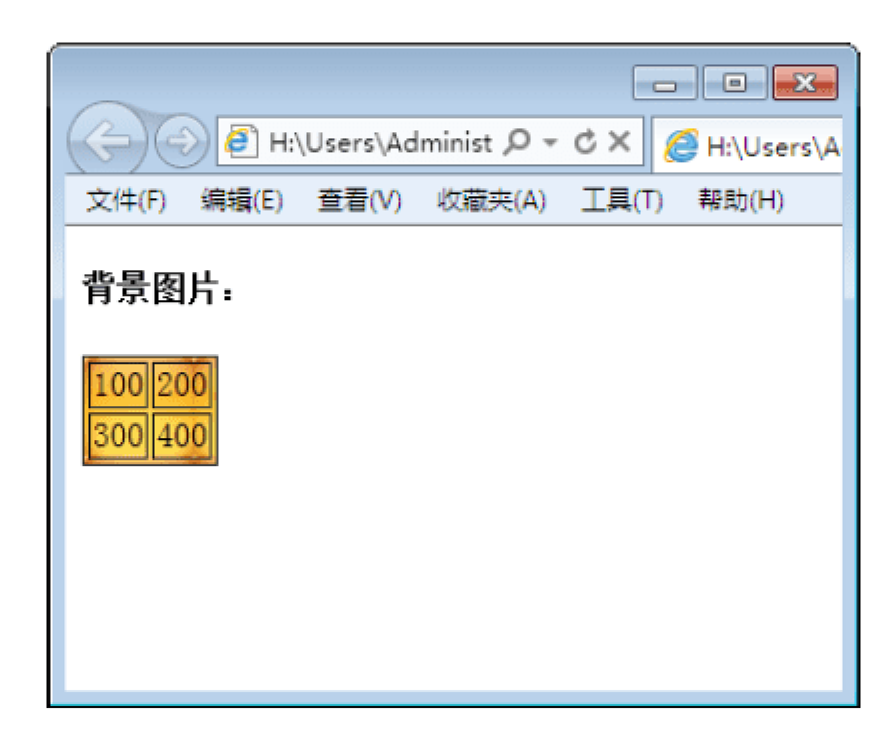


图 6-9　为表格添加背景图片

6.3.4　设置单元格背景

除了可以为表格设置背景外，还可以为单元格设置背景。下面的实例为单元格添加背景。

【例 6.9】（实例文件：ch06\6.9.html）

```
<html>
<body>
<h4>单元格背景</h4>
<table border="1">
<tr>
   <td bgcolor="red">100000</td>
   <td>200000</td>
</tr>
<tr>
   <td background="images/1.gif">200000</td>
   <td>300000</td>
</tr>
</table>
</body>
</html>
```

在 IE 9.0 中预览网页效果如图 6-10 所示。

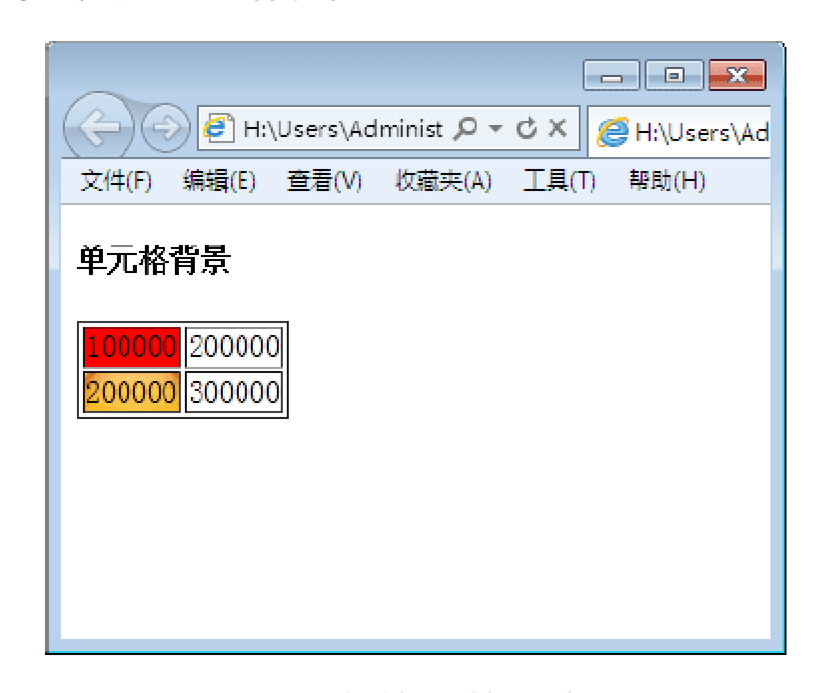


图 6-10　为单元格添加背景

6.4　综合实例——制作计算机报价表

下面，利用本章的表格知识，制作如图 6-11 所示的计算机报价表。

型号	类型	价格	图片
宏碁（Acer）AS4552-P362G32MNCC	笔记本	￥2799	
戴尔（Dell）14VR-188	笔记本	￥3499	
联想（Lenovo）G470AH2310W42G500P7CW3(DB)-CN	笔记本	￥4149	
戴尔家用（DELL）I560SR-656	台式	￥3599	

图 6-11　计算机报价表

具体操作步骤如下。

01 新建 HTML 文档，并对其简化，如下面的代码所示。

```
<html>
<head>
<title>完整表格标记</title>
</head>
<body>
</body>
</html>
```

02 保存 HTML 文件，选择相应的保存位置，文件名为“计算机报价表.html”。

03 在 HTML 文档的 body 部分增加表格及内容，如下面的代码所示。

```
<table>
    <caption>计算机报价单</caption>
     <tr>
        <th>型号</th>
        <th>类型</th>
        <th>价格</th>
        <th>图片</th>
     </tr>
     <tr>
```

```
            <td>宏碁 (Acer) AS4552-P362G32MNCC</td>
            <td>笔记本</td>
            <td>￥2799</td>
            <td><img src="images/Acer.jpg" width="120" height="120"></td>
        </tr>
        <tr>
            <td>戴尔 (Dell) 14VR-188</td><td>笔记本</td>
            <td>￥3499</td>
            <td><img src="images/Dell.jpg" width="120" height="120"></td>
        </tr>
         <tr>
            <td>联想 (Lenovo) G470AH2310W42G500P7CW3(DB)-CN</td>
            <td>笔记本</td>
            <td>￥4149</td>
            <td><img src="images/Lenovo.jpg" width="120" height="120"></td>
        </tr>
        <tr>
            <td>戴尔家用 (DELL)   I560SR-656</td>
            <td>台式</td>
            <td>￥3599</td>
            <td><img src="images/DellT.jpg" width="120" height="120"></td>
        </tr>
    </table>
```

利用 caption 标记制作表格的标题，<th>代替<td>作为标题行单元格。可以将图片放在单元格内，即在<td>标记内使用<img>标记。

04 在 HTML 文档的 head 部分，增加 CSS 样式，为表格增加边框及相应的修饰，代码如下。

```
<style>
table{
    /*表格增加线宽为 3 的橙色实线边框*/
    border:3px solid #F60;
}
caption{
    /*表格标题字号 36*/
    font-size:36px;
}
th,td{
    /*表格单元格（th、td）增加边线*/
    border:1px solid #F90;
}
```

```
</style>
```

05 保存网页后，即可查看最终效果。

6.5 专家解惑

1．在 Dreamweaver CS5.5 中如何选择多个单元格？

答：在 Dreamweaver CS5.5 中选择单元格的操作类似于文字处理工具 Word，按下鼠标左键拖动鼠标，经过的单元格都会被选中。按下 Ctrl 键，单击某个单元格，该单元格将会被选中，这些单元格可以是连续的，也可以是不连续的。在需要选择区域的开头单元格中单击，按下 SHIFT 键，在区域的末尾单元格中单击，开头和结尾单元格组成的区域内的所有单击格将会被选中。

2．表格除了显示数据，还可以进行布局，为何不使用表格进行布局？

答：在互联网刚刚开始普及时，网页非常简单，形式也非常单调，当时美国设计师 David Siegel 发明了使用表格布局，风靡全球。在表格布局的页面中，表格不但需要显示内容，还要控制页面的外观及显示位置，导致页面代码过多，结构与内容无法分离。这样就给网站的后期维护和很多其他方面带来了麻烦。

3．使用<thead>、<tbody>和<tfoot>标记对行进行分组的意义何在？

答：在 HTML 文档中增加<thead>、<tbody>和<tfoot>标记虽然从外观上不能看出任何变化，但是它们却使文档的结构更加清晰。另外，还有一个更重要的意义，即方便使用 CSS 样式对表格的各个部分进行修饰，从而制作出更炫的表格。

第 7 章

网页表单的设计

在网页中，表单的作用比较重要，主要负责采集浏览者的相关数据，例如常见的注册表、调查表和留言表等。

7.1 表单简介

表单是页面中用来提供用户交互的主要方法。一个表单有 3 个基本组成部分。

- 表单标签：这里面包含了处理表单数据所用 CGI 程序的 URL 以及数据提交到服务器的方法。
- 表单域：表单域包含文本框、密码框、隐藏域、多行文本框、复选框、单选框、下拉选择框和文件上传框等。
- 表单按钮：表单按钮包括提交按钮、复位按钮和一般按钮；用于将数据传送到服务器上的 CGI 脚本或者取消输入，还可以用表单按钮来控制其他定义了处理脚本的处理工作。

在 HTML 中，一般将表单设计在 HTML 文档中，当用户填写完信息后进行提交操作，于是表单的内容就从客户端的浏览器传送到服务器上，经过服务器上程序处理后，再将用户所需信息传送回客户端的浏览器上，这样网页就具有了交互性。

在网页中，最常见的表单形式主要包括文本框、单选按钮、复选框、按钮、下拉列表等。图 7-1 所示为一个网页应用表单之后的页面。

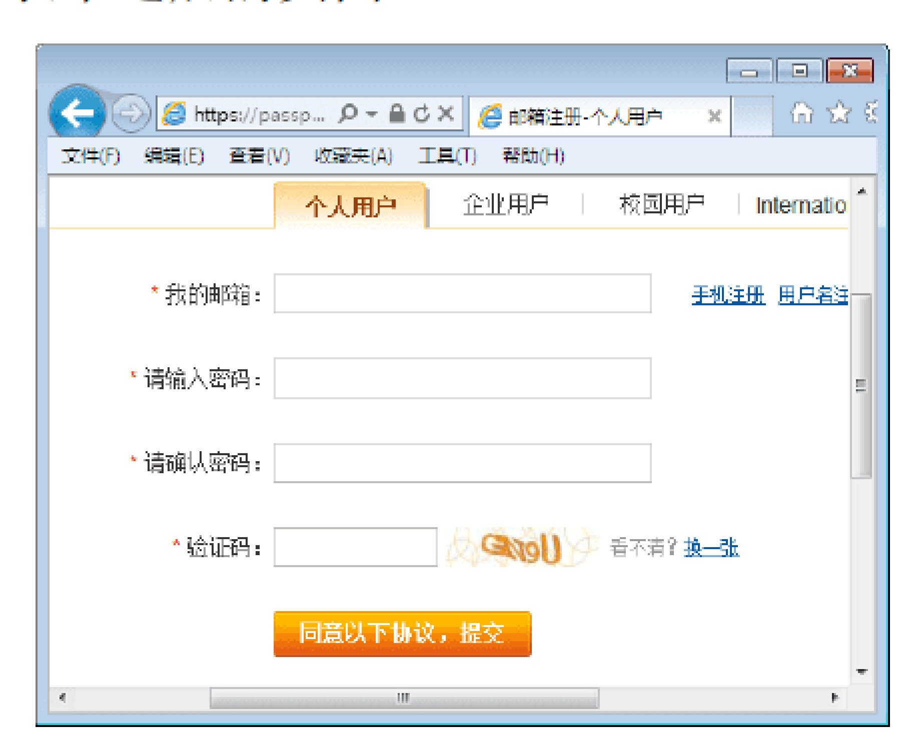


图 7-1　邮箱注册页面

7.2 表单基本元素

表单元素是能够让用户在表单中输入信息的元素。常见的有文本框、密码框、下拉菜单、单选框、复选框等。本章节主要讲述表单基本元素的使用方法和技巧。

7.2.1 表单标签

表单主要用于收集网页上浏览者的相关信息，其标签为<form>、</form>。表单的基本语法格式如下。

```
<form action="url" method="get|post" enctype="mime">
</form >
```

其中，action=url 指定处理提交表单的格式，它可以是一个 URL 地址或一个电子邮件地址。method=get 或 post 指明提交表单的 HTTP 方法。enctype=mine 指明用来把表单提交给服务器时的互联网媒体形式。

表单是一个能够包含表单元素的区域。通过添加不同的表单元素，将显示不同的效果。

【例 7.1】（实例文件：ch07\7.1.html）

```
<html>
<body>
<form>
下面是输入用户登录信息
<br>
用户名称
<input type="text" name="user">
<br>
用户密码
<input type="password" name="password">
<br>
<input type="submit" value="登录">
</form>
</body>
</html>
```

在 IE 9.0 中浏览效果如图 7-2 所示，可以看到用户登录信息页面。

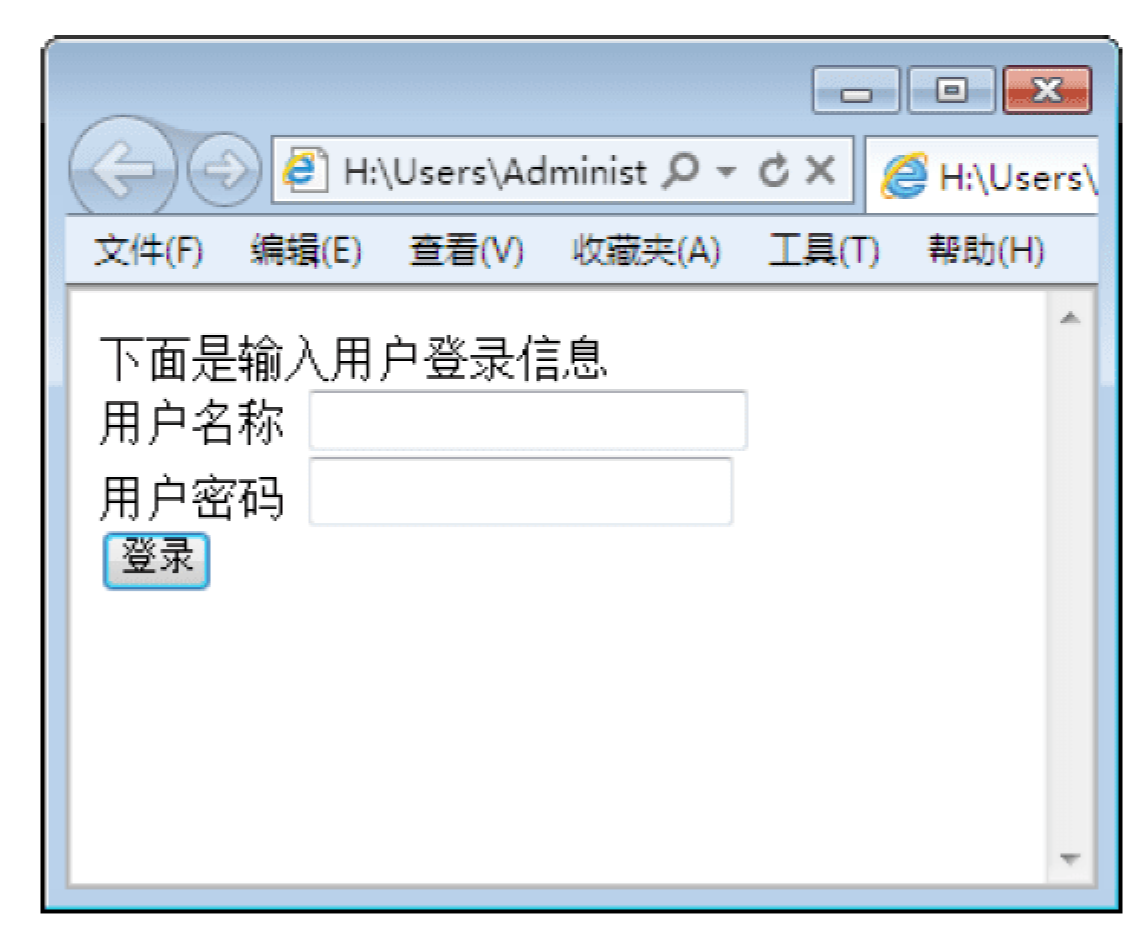


图 7-2　用户登录窗口

7.2.2　文字字段

在网页中输入文字字段的方法主要有两种，一种是单行文本输入框；一种是多行文本输入框。

1．单行文本输入框（text）

文本框是一种让访问者自己输入内容的表单对象，通常用来填写单个字或者简短的回答，例如用户姓名和地址等。代码格式如下。

```
<input type="text" name="..." size="..." maxlength="..." value="...">
```

其中，type="text"定义单行文本输入框，name 属性定义文本框的名称，要保证数据的准确采集，必须定义一个独一无二的名称；size 属性定义文本框的宽度，单位是单个字符宽度；maxlength 属性定义最多输入的字符数；value 属性定义文本框的初始值。

【例 7.2】（实例文件：ch07\7.2.html）

```
<html>
<head><title>输入用户的姓名</title></head>
<body>
<form>
请输入您的姓名：
<input type="text" name="yourname" size="20" maxlength="15">
请输入您的地址：
<input type="text" name="youradr" size="20" maxlength="15">
</form>
</body>
</html>
```

在 IE 9.0 中浏览效果如图 7-3 所示，可以看到两个单行文本输入框。

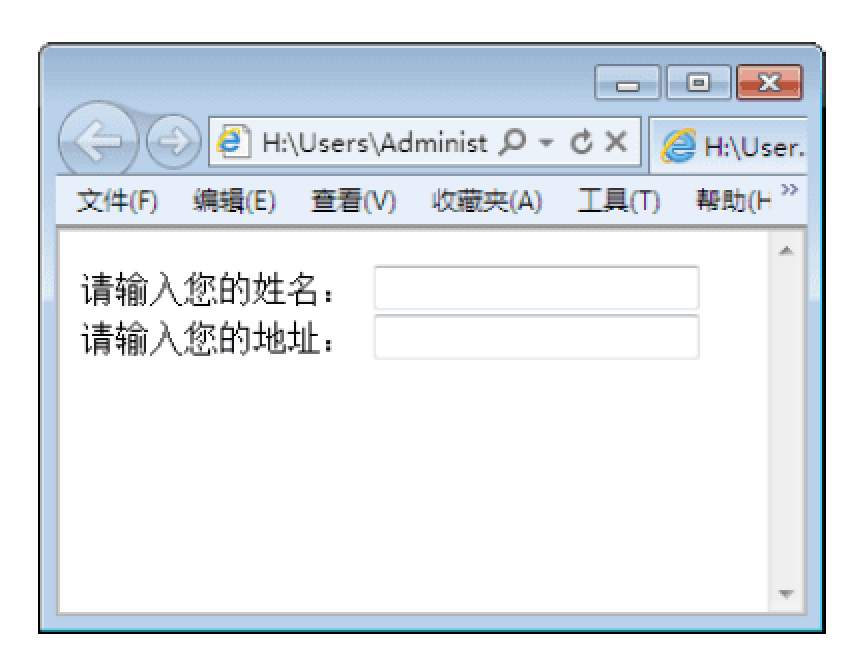


图 7-3 单行文本输入框

2. 多行文本输入框（textarea）

多行输入框（textarea）主要用于输入较长的文本信息。代码格式如下。

```
<textarea name="..." cols="..." rows="..." wrap="..."></textarea >
```

其中，name 属性定义多行文本框的名称，要保证数据的准确采集，必须定义一个独一无二的名称；cols 属性定义多行文本框的宽度，单位是单个字符宽度；rows 属性定义多行文本框的高度，单位是单个字符宽度。wrap 属性定义输入内容大于文本域时显示的方式。

【例 7.3】（实例文件：ch07\7.3.html）

```
<html>
<head><title>多行文本输入</title></head>
<body>
<form>
请输入您最近的工作情况<br>
<textarea name="yourworks" cols ="50" rows = "5"></textarea>
<br>
<input type="submit" value="提交">
</form>
</body>
</html>
```

在 IE 9.0 中浏览效果如图 7-4 所示，可以看到多行文本输入框。

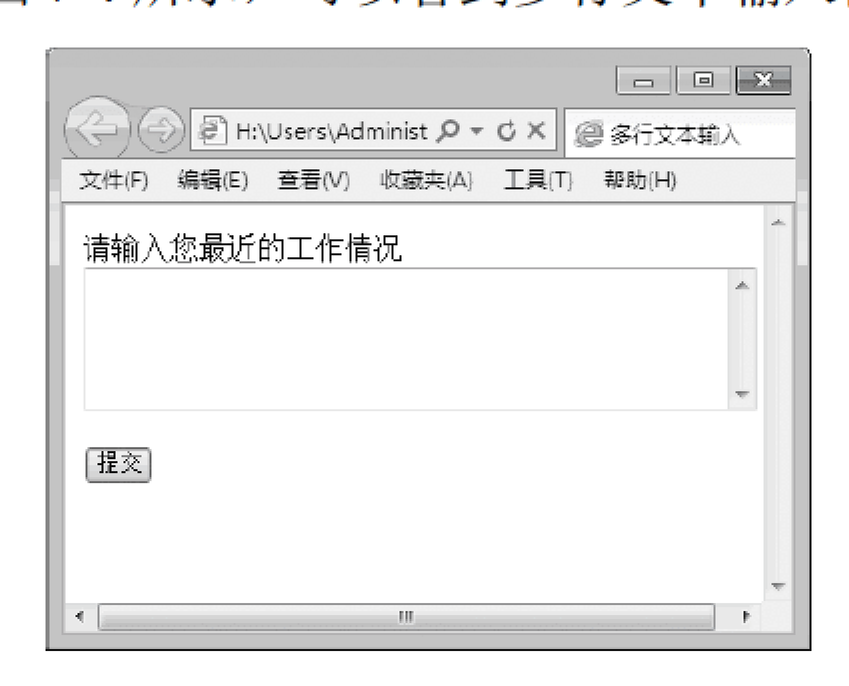


图 7-4 多行文本输入框

7.2.3　密码域

密码输入框是一种特殊的文本域，主要用于输入一些保密信息。当网页浏览者输入文本时，显示的是黑点或者其他符号，这样就增加了输入文本的安全性。代码格式如下。

```
<input type="password" name="..." size="..." maxlength="...">
```

其中 type="password"定义密码框；name 属性定义密码框的名称，要保证唯一性；size 属性定义密码框的宽度，单位是单个字符宽度；maxlength 属性定义最多输入的字符数。

【例 7.4】（实例文件：ch07\7.4.html）

```
<html>
<head><title>输入用户姓名和密码 </title></head>
<body>
<form >
用户姓名：
<input type="text" name="yourname">
<br>
登录密码：
<input type="password" name="yourpw"><br>
</form>
</body>
</html>
```

在 IE 9.0 中浏览效果如图 7-5 所示，输入密码时可以看到密码以黑点的形式显示。

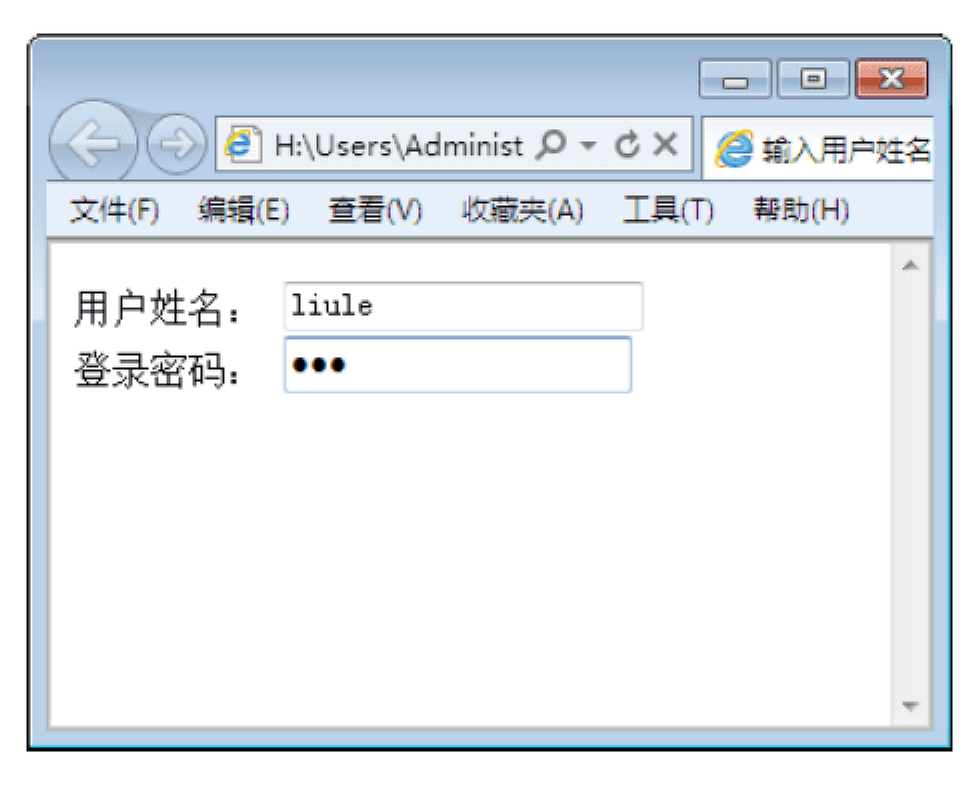


图 7-5　密码输入框

7.2.4　单选按钮

单选按钮主要是让网页浏览者在一组选项里选择一个。代码格式如下。

```
<input type="radio" name=" " value = " ">
```

其中 type="radio"定义单选按钮，name 属性定义单选按钮的名称，单选按钮都是以组为单

位使用的，在同一组中的单选项都必须使用同一个名称；value 属性定义单选按钮的值，在同一组中，它们的域值必须是不同的。

【例 7.5】（实例文件：ch07\7.5.html）

```
<html>
<head><title>选择感兴趣的图书</title></head>
<body>
<form >
请选择您感兴趣的图书类型：
<br>
<input type="radio" name="book" value = "Book1">网站编程<br>
<input type="radio" name="book" value = "Book2">办公软件<br>
<input type="radio" name="book" value = "Book3">设计软件<br>
<input type="radio" name="book" value = "Book4">网络管理<br>
<input type="radio" name="book" value = "Book5">黑客攻防<br>
</form>
</body>
</html>
```

在 IE 9.0 中浏览效果如图 7-6 所示，可以看到 5 个单选按钮，用户只能选择其中一个单选按钮。

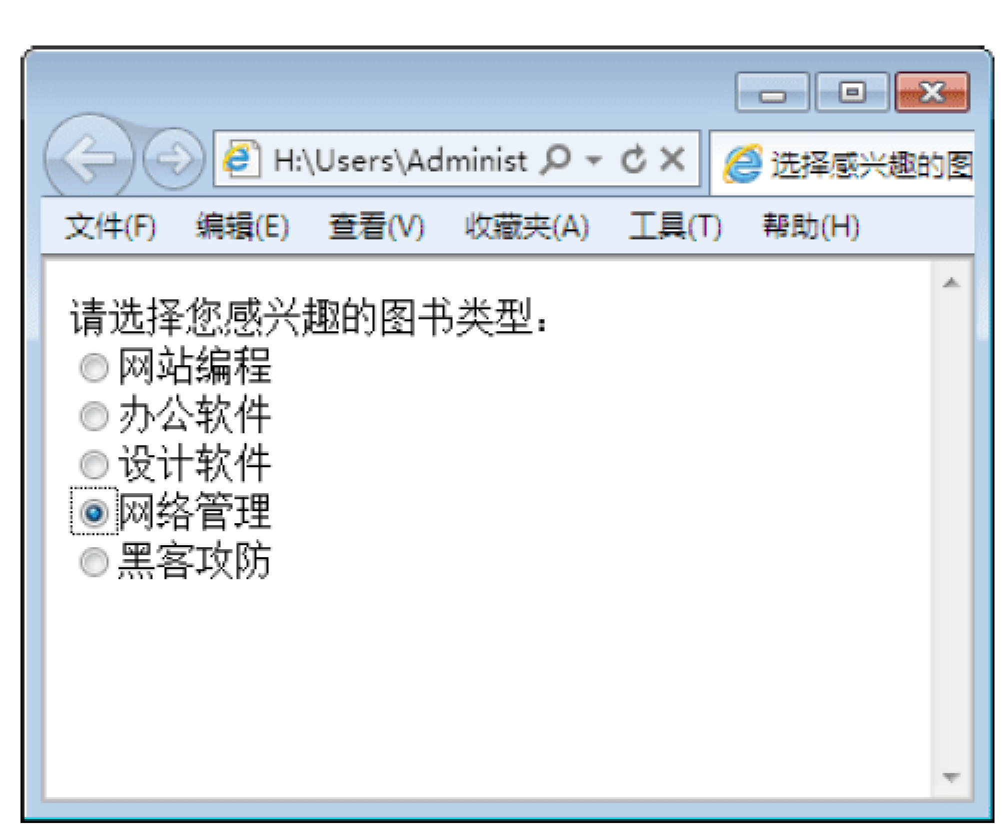


图 7-6 单选按钮

7.2.5 复选框

复选框主要是让网页浏览者在一组选项里同时选择多个选项。每个复选框都是一个独立的元素，都必须有一个唯一的名称。代码格式如下。

```
<input type="checkbox" name=" " value ="">
```

其中 type="checkbox"定义复选框；name 属性定义复选框的名称，在同一组中的复选框都必须用同一个名称；value 属性定义复选框的值。

【例 7.6】（实例文件：ch07\7.6.html）

```
<html>
<head><title>选择感兴趣的图书</title></head>
<body>
<form >
请选择您感兴趣的图书类型：<br>
<input type="checkbox" name="book" value = "Book1">网站编程<br>
<input type="checkbox" name="book" value = "Book2">办公软件<br>
<input type="checkbox" name="book" value = "Book3">设计软件<br>
<input type="checkbox" name="book" value = "Book4">网络管理<br>
<input type="checkbox" name="book" value = "Book5" checked>黑客攻防<br>
</form>
</body>
</html>
```



checked 属性主要是设置默认选中项。

在 IE 9.0 中浏览效果如图 7-7 所示，即可看到 5 个复选框，其中【黑客攻防】复选框默认被选中。

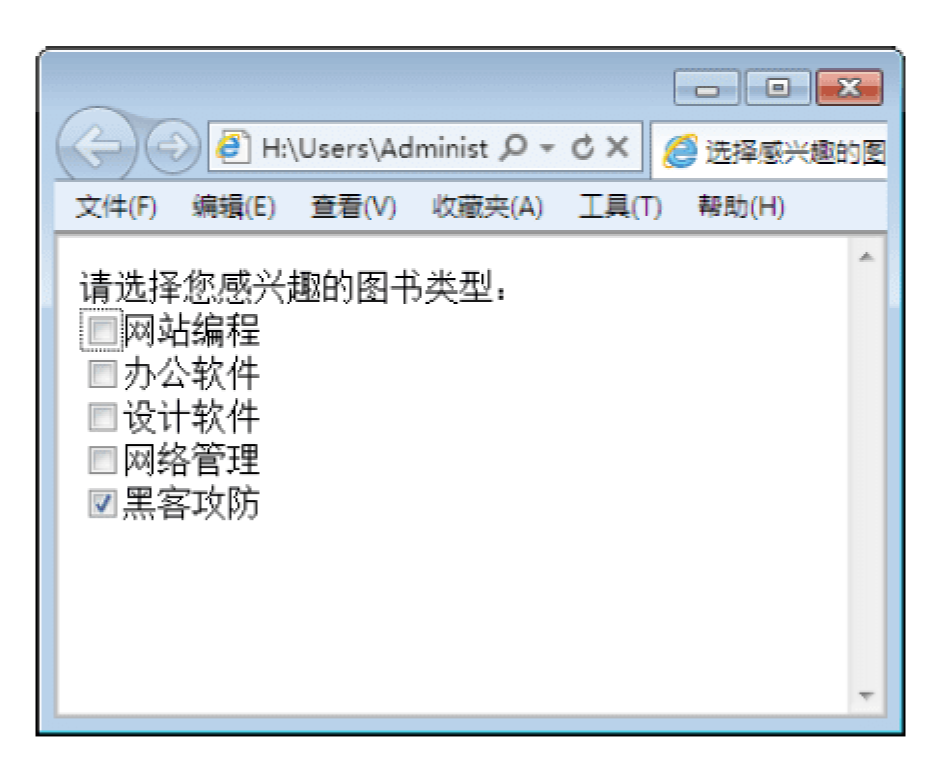


图 7-7 复选框的效果

7.2.6 下拉菜单

使用 select 元素可创建单选或多选下拉菜单。

【例 7.7】（实例文件：ch07\7.7.html）

```
<html>
<body>
<select>
  <option>列表 1</option>
  <option>列表 2</option>
  <option>列表 3</option>
```

```
    <option>列表 4</option>
  </select>
  </body>
  </html>
```

在 IE 9.0 中浏览效果如图 7-8 所示，单击列表框右侧的下三角按钮，在弹出的下拉列表中进行选择。

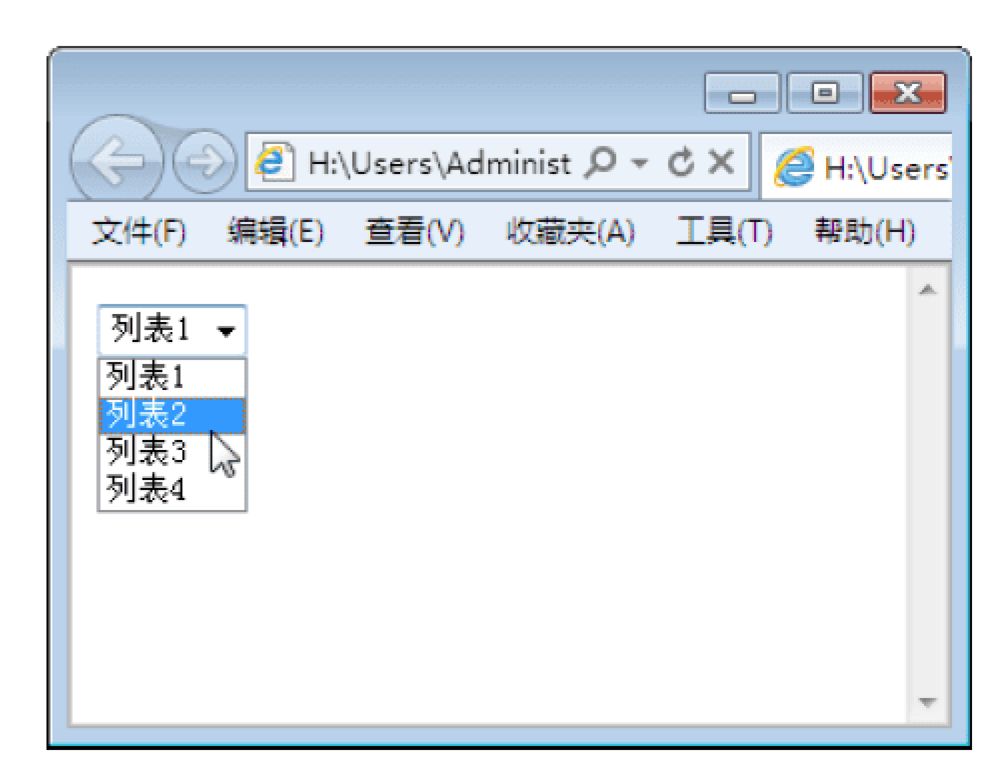


图 7-8 下拉菜单列表

7.2.7 列表框

列表框主要用于在有限的空间里设置多个选项。列表框既可以用作单选，也可以用作复选。代码格式如下。

```
<select name="..." size="..." multiple>
<option value="..." selected>
...
</option>
  ...
</select>
```

其中 size 属性定义下拉选择框的行数；name 属性定义下拉选择框的名称；multiple 属性表示可以多选，如果不设置本属性，那么只能单选；value 属性定义选择项的值；selected 属性表示默认已经选择本选项。

【例 7.8】（实例文件：ch07\7.8.html）

```
<html>
<head><title>选择感兴趣的图书</title></head>
<body>
<form>
请选择您感兴趣的图书类型：<br>
<select name="fruit" size = "3" multiple>
<option value="Book1">网站编程
```

```
<option value="Book2">办公软件
<option value="Book3">设计软件
<option value="Book4">网络管理
<option value="Book5">黑客攻防
</select>
</form>
</body>
</html>
```

在 IE 9.0 中浏览效果如图 7-9 所示，即可看到列表选择框，其中显示为 3 行选项，用户可以按住 Ctrl 键，选择多个选项。

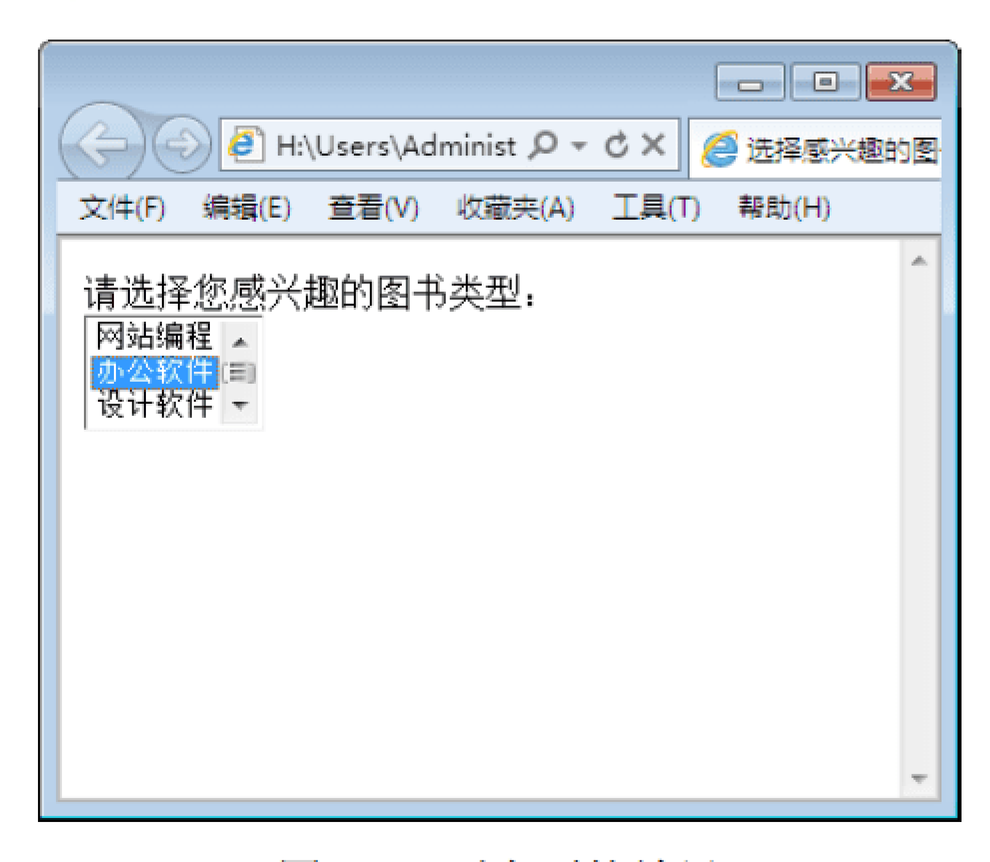


图 7-9　列表项的效果

7.2.8　普通按钮

普通按钮用来控制其他定义了处理脚本的处理工作。代码格式如下。

```
<input type="button" name="..." value="..." onClick="...">
```

其中 type="button"定义普通按钮；name 属性定义普通按钮的名称；value 属性定义按钮的显示文字；onClick 属性表示单击行为，也可以是其他的事件，通过指定脚本函数来定义按钮的行为。

【例 7.9】（实例文件：ch07\7.9.html）

```
<html>
<body>
<form>
单击下面的按钮，把文本框 1 中的内容复制到文本框 2 中：
<br/>
文本框 1: <input type="text" id="field1" value="学习 HTML 的技巧">
<br/>
文本框 2: <input type="text" id="field2">
```

```
<br/>
<input type="button" name="..."value="单击我"onClick="document.getElementById('field2').value=document.getElementById('field1').value">
</form>
</body>
</html>
```

在 IE 9.0 中浏览效果如图 7-10 所示，单击【单击我】按钮，即可将文本框 1 中内容复制到文本框 2 中。

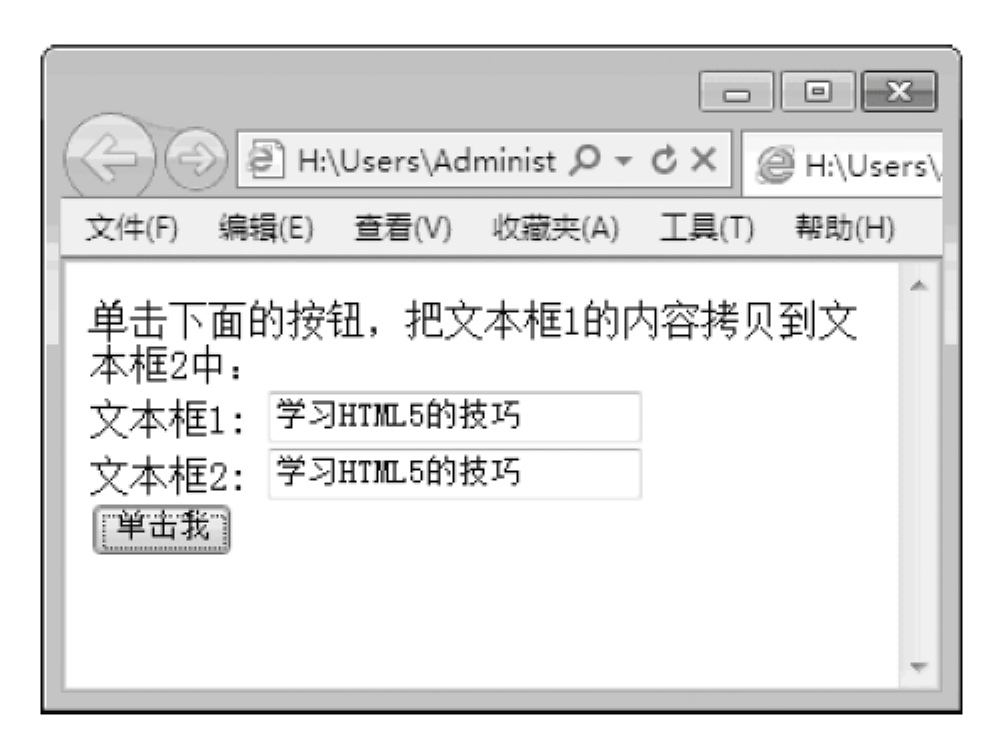


图 7-10 单击按钮后的复制效果

7.2.9 提交按钮

提交按钮用来将输入的信息提交到服务器。代码格式如下。

```
<input type="submit" name="..." value="...">
```

其中 type="submit"定义提交按钮；name 属性定义提交按钮的名称；value 属性定义按钮的显示文字。通过提交按钮可以将表单里的信息提交给表单里 action 所指向的文件。

【例 7.10】（实例文件：ch07\7.10.html）

```
<html>
<head><title>输入用户名信息</title></head>
<body>
<form  action="http://www.yinhangit.com/yonghu.asp" method="get">
请输入你的姓名：
<input type="text" name="yourname">
<br>
请输入你的住址：
<input type="text" name="youradr">
<br>
请输入你的单位：
<input type="text" name="yourcom">
<br>
```

```
请输入你的联系方式：
<input type="text" name="yourcom">
<br>
<input type="submit" value="提交">
</form>
</body>
</html>
```

在 IE 9.0 中浏览效果如图 7-11 所示，输入内容后单击【提交】按钮，即可实现将表单中的数据发送到制定的文件。

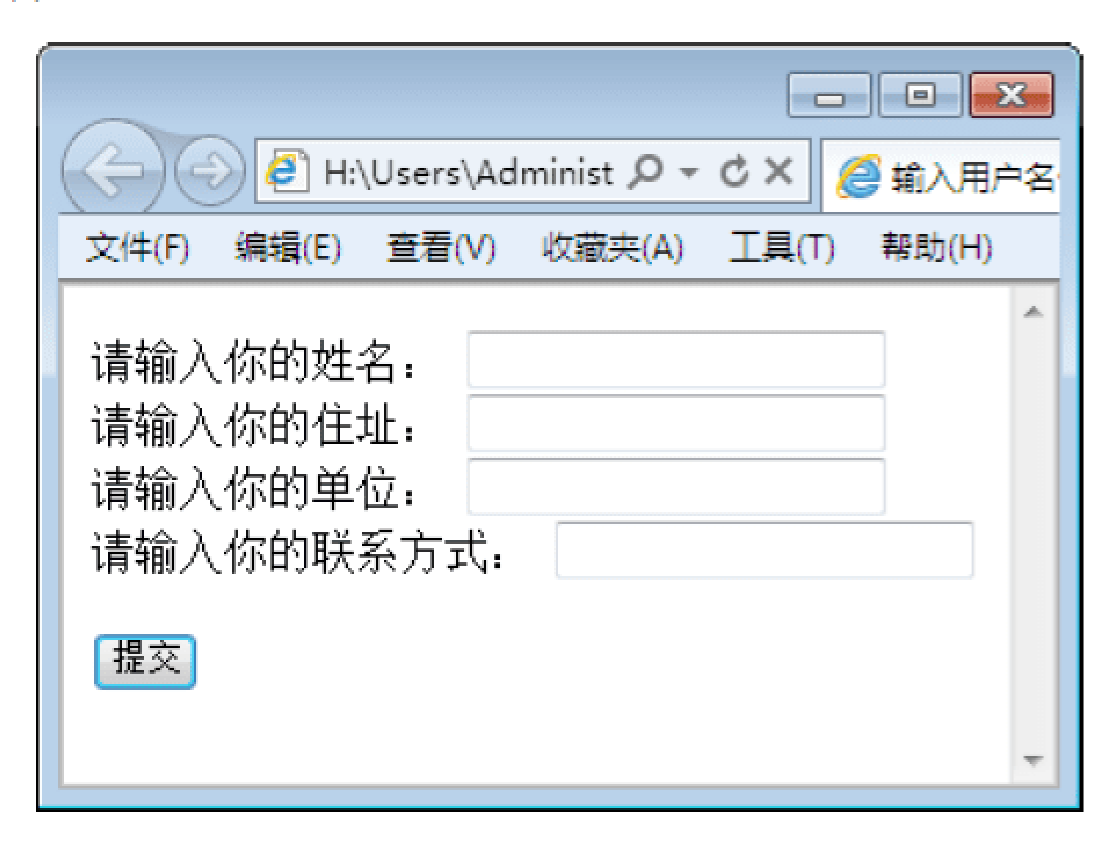


图 7-11　提交按钮

7.2.10　重置按钮

重置按钮用来重置表单中输入的信息。代码格式如下。

```
<input type="reset" name="..." value="...">
```

其中 type="reset"定义重置按钮；name 属性定义重置按钮的名称；value 属性定义按钮的显示文字。

【例 7.11】（实例文件：ch07\7.11.html）

```
<html>
<body>
<form>
请输入用户名称：
<input type='text'>
<br/>
请输入用户密码：
<input type='password'>
<br>
<input type="submit" value="登录">
<input type="reset" value="重置">
```

```
</form>
</body>
</html>
```

在 IE 9.0 中浏览效果如图 7-12 所示，输入内容后单击【重置】按钮，即可实现将表单中的数据清空的目的。

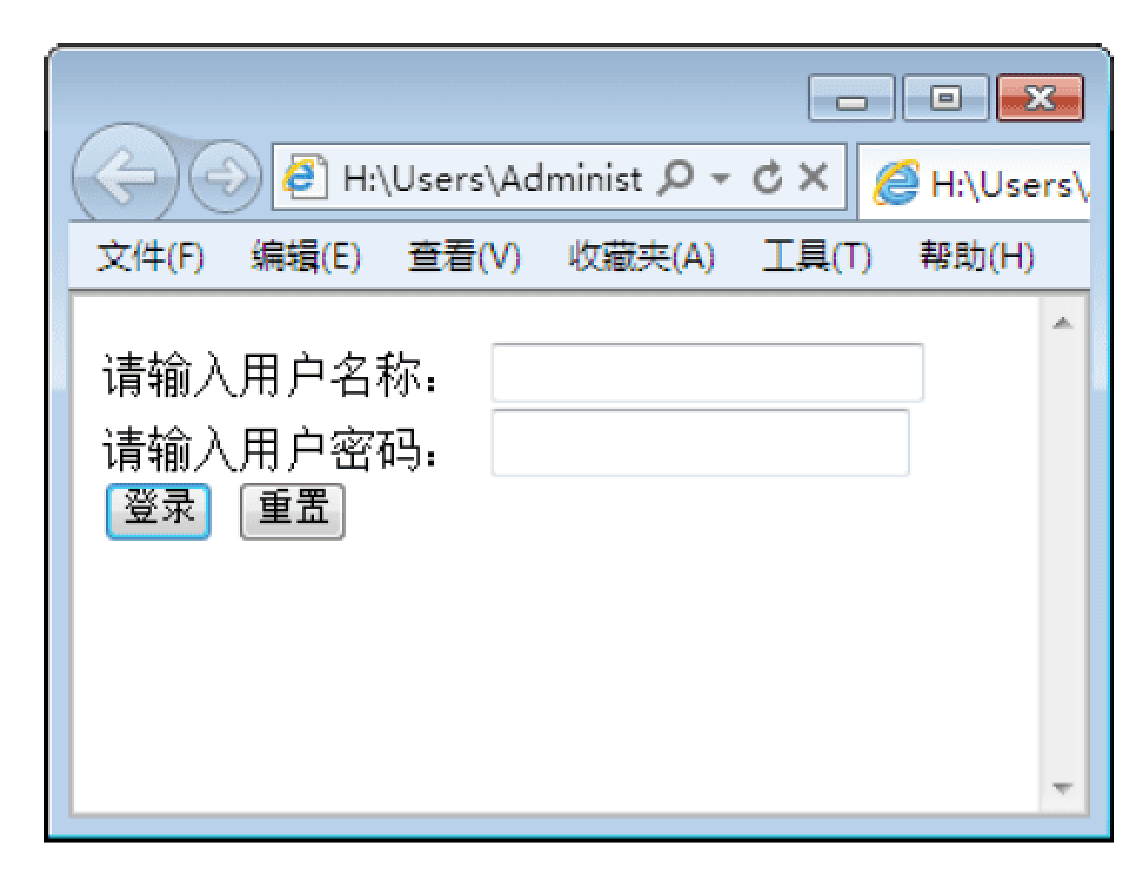


图 7-12 重置按钮

7.3 综合实战——制作淘宝注册页

本实例中，将使用一个表单内的各种元素来开发网站的简单用户意见反馈页面。

具体操作步骤如下。

01 分析需求。

反馈表单非常简单，通常包含 3 个部分，需要在页面上方给出标题，标题下方是正文部分，即表单元素，最下方是表单元素提交按钮。在设计这个页面时，需要把“用户注册”标题设置成 H1 大小，正文使用 p 来限制表单元素。

02 构建 HTML 页面，实现表单内容。

```
<html>
<head>
<title>淘宝注册页面</title>
</head>
<body>
<h1 align=center>淘宝注册页面</h1>
<form method="post" >
<p>姓    名：
<input type="text" class=txt size="12" maxlength="20" name="username" />
</p><p>性    别：
```

```
<input type="radio" value="male" />男
<input type="radio" value="female" />女
</p><p>年    龄：
<input type="text" class=txt name="age"   />
</p>
<p>联系电话：
<input type="text" class=txt name="tel" />
</p><p>电子邮件：
<input type="text" class=txt name="email" />
</p><p>联系地址：
<input type="text"   class=txt name="address" />
</p>
<p>
请输入您对网站的建议<br>
<textarea name="yourworks" cols ="50" rows = "5"></textarea>
<br>
<input type="submit" name="submit" value="提交"/>
<input type="reset" name="reset" value="清除" />
</p>
</form>
</body>
</html>
```

在 IE 9.0 中浏览效果如图 7-13 所示，可以看到创建了一个用户反馈表单，包含标题“淘宝注册页面”、“姓名”、“性别”、“年龄”、“联系电话”、“电子邮件”、“联系地址”输入框和“提交”、“清除”按钮。

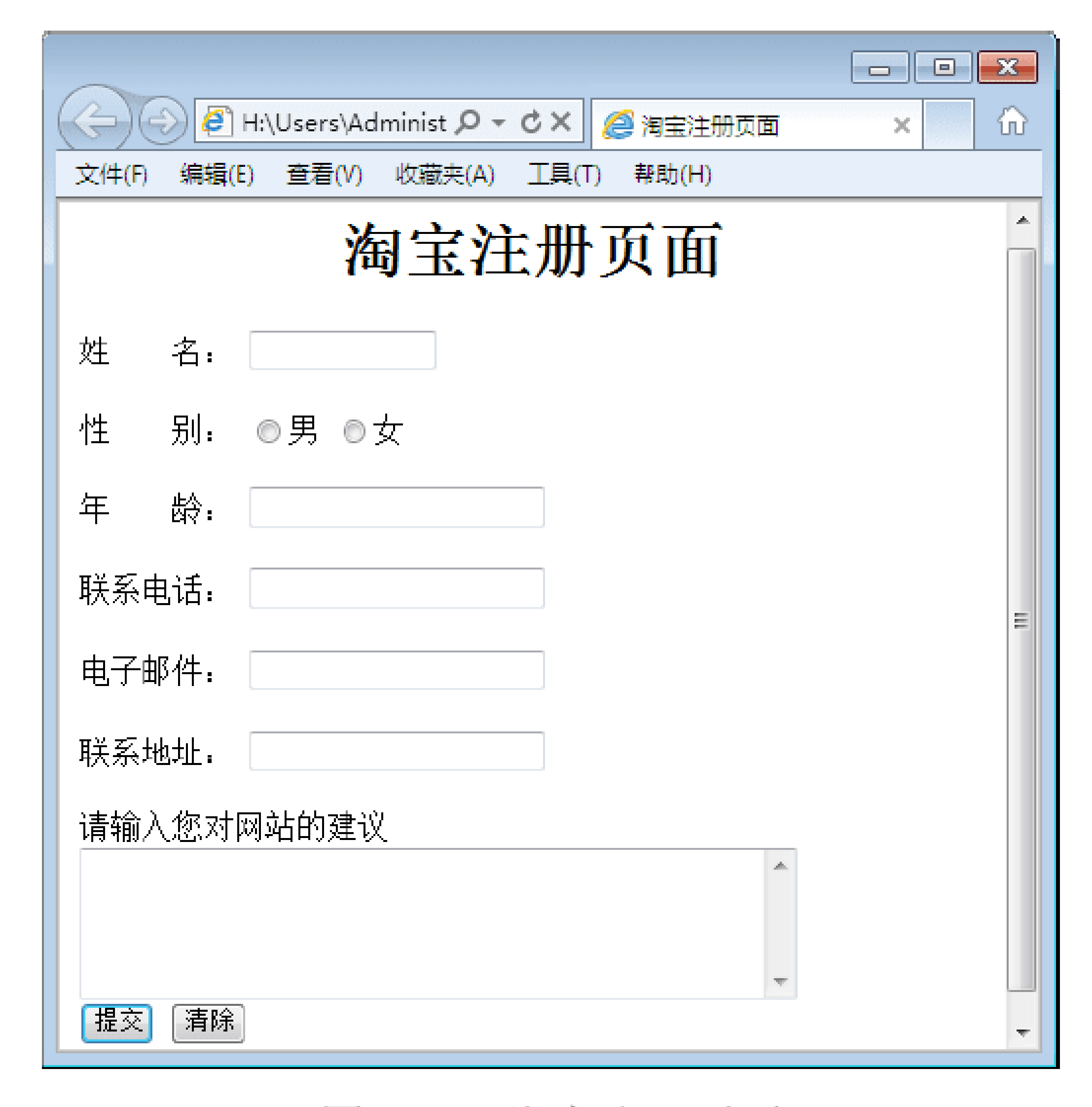


图 7-13　淘宝注册页面

7.4 专家解惑

1. 如何在表单中实现文件上传框？

在 HTML 语言中，使用 file 属性实现文件上传框。语法格式为：<input type="file" name="..." size=" " maxlength=" ">。其中 type="file"定义为文件上传框，name 属性为文件上传框的名称，size 属性定义文件上传框的宽度，单位是单个字符宽度；maxlength 属性定义最多输入的字符数。文件上传框的显示效果如图 7-14 所示。

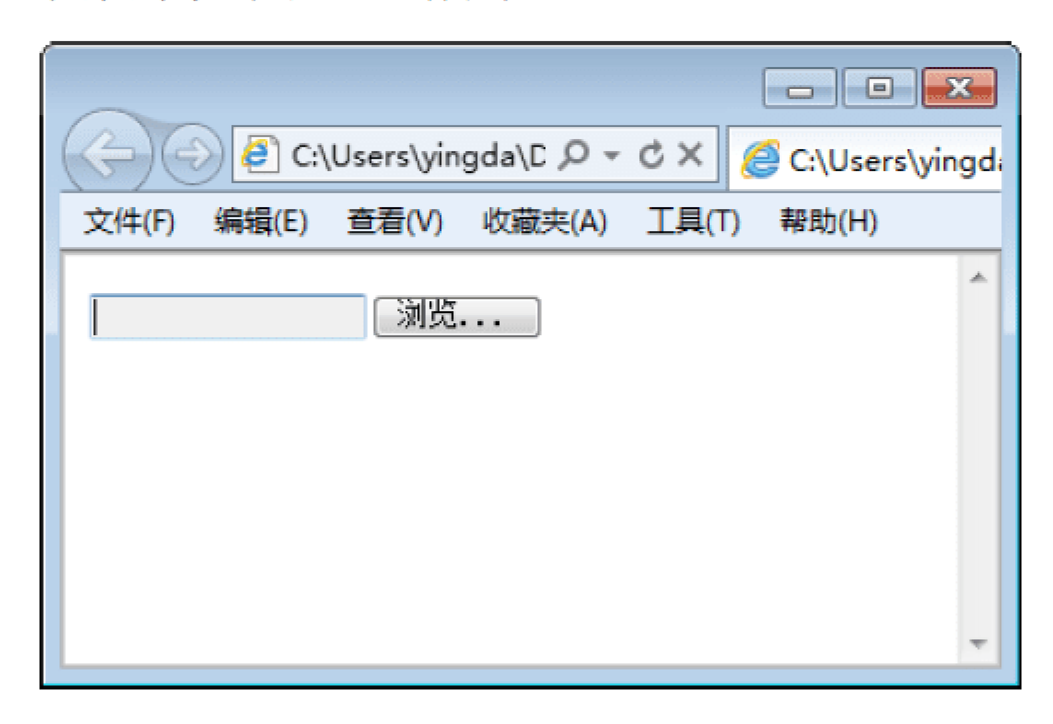


图 7-14 文件上传框

2. 制作的单选框为什么可以同时选中多个？

此时用户需要检查单选框的名称，要保证同一组中的单选框名称相同，这样才能保证单选框只能同时选中其中一个。

第 8 章

使用HTML创建框架

框架通常用来定义页面的导航区域和内容区域，使用框架最常见的情况就是一个框架显示包含导航栏的文档，而另一个框架显示包含内容的文档。框架是网页中最常用的页面设计方式，很多网站都使用了框架技术。

8.1 窗口框架简介

框架的最主要功能是用来“分割”页面窗口，使每个“小窗口”能显示不同的 HTML 文件，这样的页面结构就称为框架结构的页面，而这些“小窗口”就称为框架的“窗口”。

8.1.1 什么是框架

框架主要用于在一个浏览器窗口中显示多个 HTML 文档内容，通过构建这些文档之间的相互关系，实现文档导航、浏览以及操作等目的。框架可以显示与浏览器窗口其余部分中显示内容无关的 HTML 文档。

框架网页有以下几个优点。

- 可以很好地保持网站风格的统一。由于框架页面中导航的部分是同一个网页，因此整体风格统一。
- 便于浏览者访问。框架网页中导航部分是固定的，不需要滚动条，便于浏览者访问阅读。
- 可以提高网页的制作效率。可以把每个网页都用到的公共内容制作成单独的网页，作为框架网页的一个框架页面，这样，就不需要在每个页面中重新输入这个公共部分的内容了，可以节省时间，提高效率。
- 可以方便更新、维护网站。在更新网站时，只需修改公共部分的框架内容，使用这个框架内容的文档会自动更新，就能完成整个网站的更新修改。



框架在网站的首页面中比较常见。在一个页面中，可以使用框架的嵌套来实现网页设计中的多种需求。对框架还可以设置边框的颜色，可随意地设置框架的链接和跳转功能，还可以设置框架的行为，从而制作出更复杂的页面。

8.1.2　框架的基本结构

框架主要包含两个部分，一个是框架集，另一个是具体的框架文件。

框架集主要用来定义 HTML 文件为框架模式，并设定视窗如何分割文件。通俗一点说，框架集就是存放框架结构的文件，也是访问框架文件的入口文件。如果网页由左右两个框架组成，那么除了左右两个网页文件之外，还有一个总的框架集文件。

框架是页面中定义的每一个显示区域，也可以说一个窗口就是一个框架。框架页面中最基本的内容就是框架集文件，它是整个框架页面的导航文件，其基本语法如下。

```
<html>
<head>
<title>框架页面的标题</title>
</head>
<frameset>
    <frame>
    <frame>
    ……
</frameset>
</html>
```

从上面的语法结构可以看到，在使用框架的页面中，<body>主体标记被框架标记<frameset>所代替。而对于框架页面中包含的每一个框架，都是通过<frame>标记来定义的。



不能将<body></body>标签与<frameset></frameset>标签同时使用！不过，如果要添加包含一段文本的<noframes>标签，就必须将这段文字嵌套于<body></body>标签内。

8.2　设置框架集的基本属性

在了解了什么是窗口框架之后，下面就来介绍如何设置框架集的基本属性，如分割框架窗口的方式、设置框架边框的宽度和颜色等。

8.2.1　水平分割窗口

水平分割窗口是将页面沿水平方向切割，也就是将页面分成上下排列的多个窗口。其语法结构如下。

```
<frameset rows="框架窗口的高度,框架窗口的高度,……">
  <frame>
  <frame>
  ……
</frameset>
```

在上述语法中，rows 中可以取多个值，每个值表示一个框架窗口的水平宽度，它的单位可以是像素，也可以是占浏览器的百分比。

注意：一般设定了几个 rows 的值，就需要有几个框架，即需要有相应数量的<frame>参数。

【例 8.1】（实例文件：ch08\8.1.html）

```
<html>
<head>
<title>水平分割窗口</title>
</head>
<frameset rows="30%,40%,30%">
   <frame>
   <frame>
   <frame>
</frameset>
</html>
```

运行代码，可以看到页面被分割成上下 3 个窗口，其效果如图 8-1 所示。当浏览器大小变化时，框架也会随之等比例缩放。

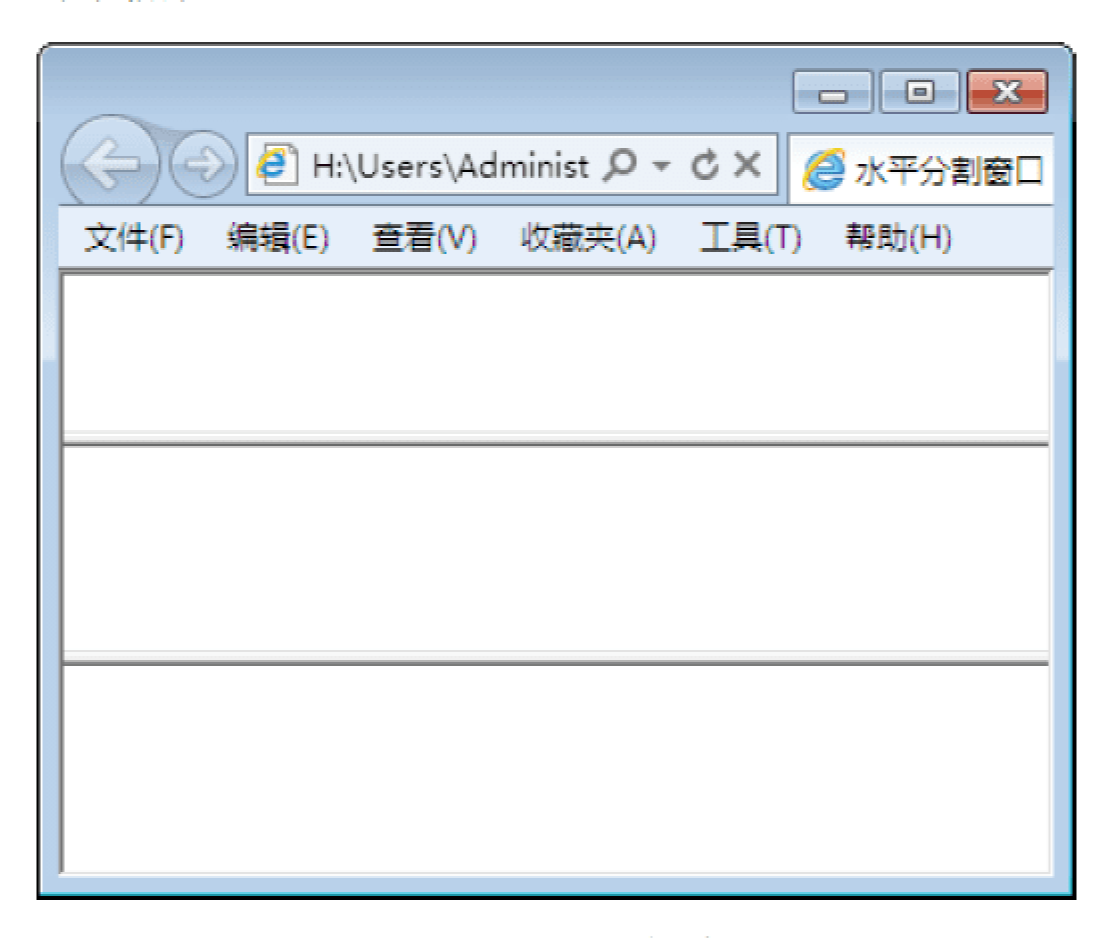


图 8-1　水平分割窗口

8.2.2　垂直分割窗口

垂直分割窗口就是将页面沿垂直方向分割成多个窗口，也就是将页面分成左右排列的多个窗口。其语法结构如下。

```
<frameset cols="框架窗口的宽度,框架窗口的宽度,……">
   <frame>
   <frame>
   ……
</frameset>
```

在上述语法中，cols 中可以取多个值，每个值表示一个框架窗口的水平宽度，它的单位可以是像素，也可以是占浏览器的百分比。与水平分割窗口相同，一般设定了几个 cols 的值，就需要有几个框架，也就是有几个<frame>参数。

【例 8.2】（实例文件：ch08\8.2.html）

```
<html>
<head>
<title>垂直分割窗口</title>
</head>
<frameset cols="25%,50%,25%">
    <frame>
    <frame>
    <frame>
</frameset>
</html>
```

运行代码，可以看到页面被分割成左右三个窗口，效果如图 8-2 所示。当浏览器大小变化时，框架也会随之等比例缩放。

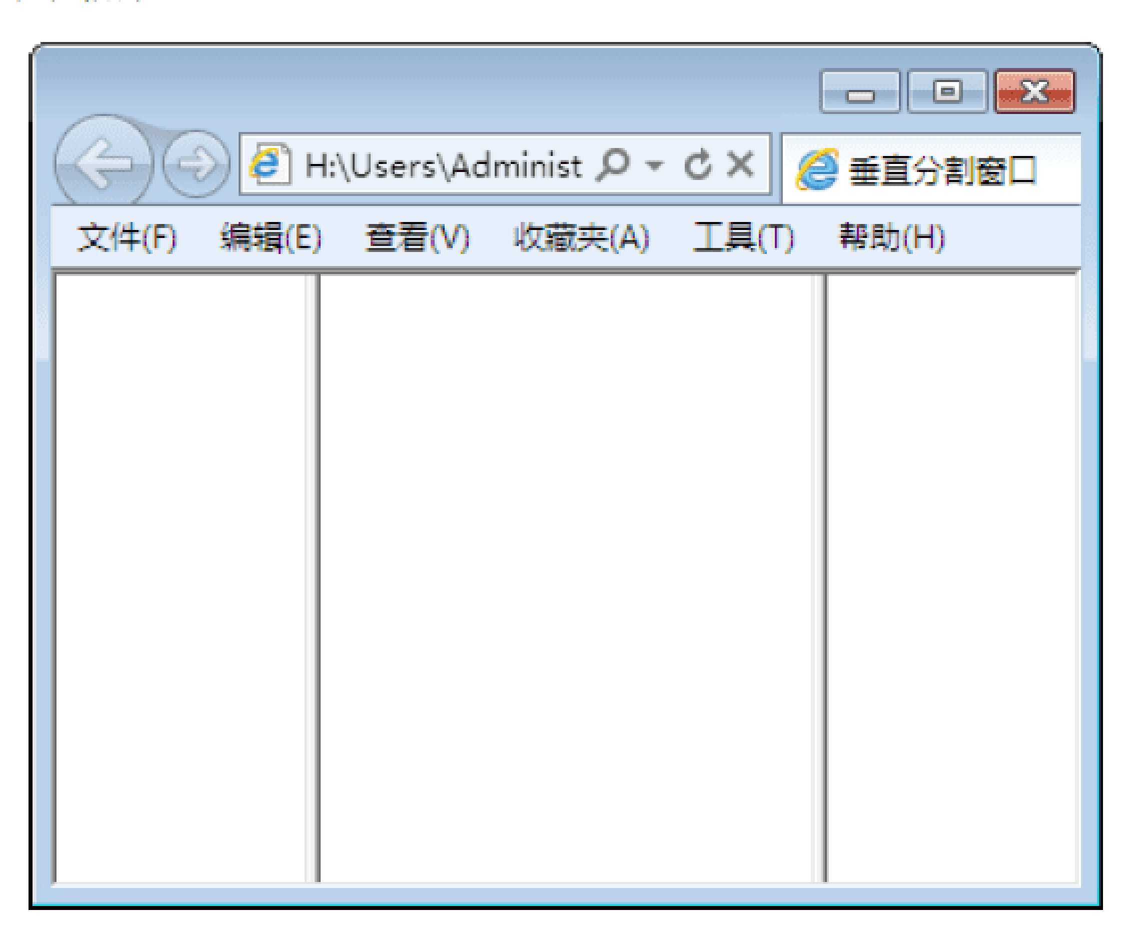


图 8-2　垂直分割窗口

8.2.3　混合分割窗口

混合分割窗口就是在一个页面中，既有水平分割的框架，又有垂直分割的框架。其语法结构如下：

```
<frameset rows="框架窗口的高度,框架窗口的高度,……">
<frame>
<frameset cols="框架窗口的宽度,框架窗口的宽度,……">
<frame>
<frame>
……
```

```
</frameset>
<frame>
……
</frameset>
```

当然，也可以先进行垂直分割，再进行水平分割。其语法结构如下：

```
<frameset cols="框架窗口的宽度,框架窗口的宽度,……">
<frame>
<frameset rows="框架窗口的高度,框架窗口的高度,……">
<frame>
<frame>
……
</frameset>
<frame>
……
</frameset>
```



在设置框架窗口时，一定要注意窗口大小的设置与窗口个数的统一。

【例 8.3】（实例文件：ch08\8.3.html）

```
<html>
<head>
<title>混合分割窗口</title>
</head>
<frameset rows="30%,70%">
<frame>
  <frameset cols="20%,55%,25%">
<frame>
<frame>
<frame>
  </frameset>
</frameset>
</html>
```

由代码可以看出，首先将页面水平分割成上下两个窗口，接着下面的框架又被垂直分割成 3 个窗口。因此下面的框架标记<frame>被框架集标记代替。运行程序，效果如图 8-3 所示。

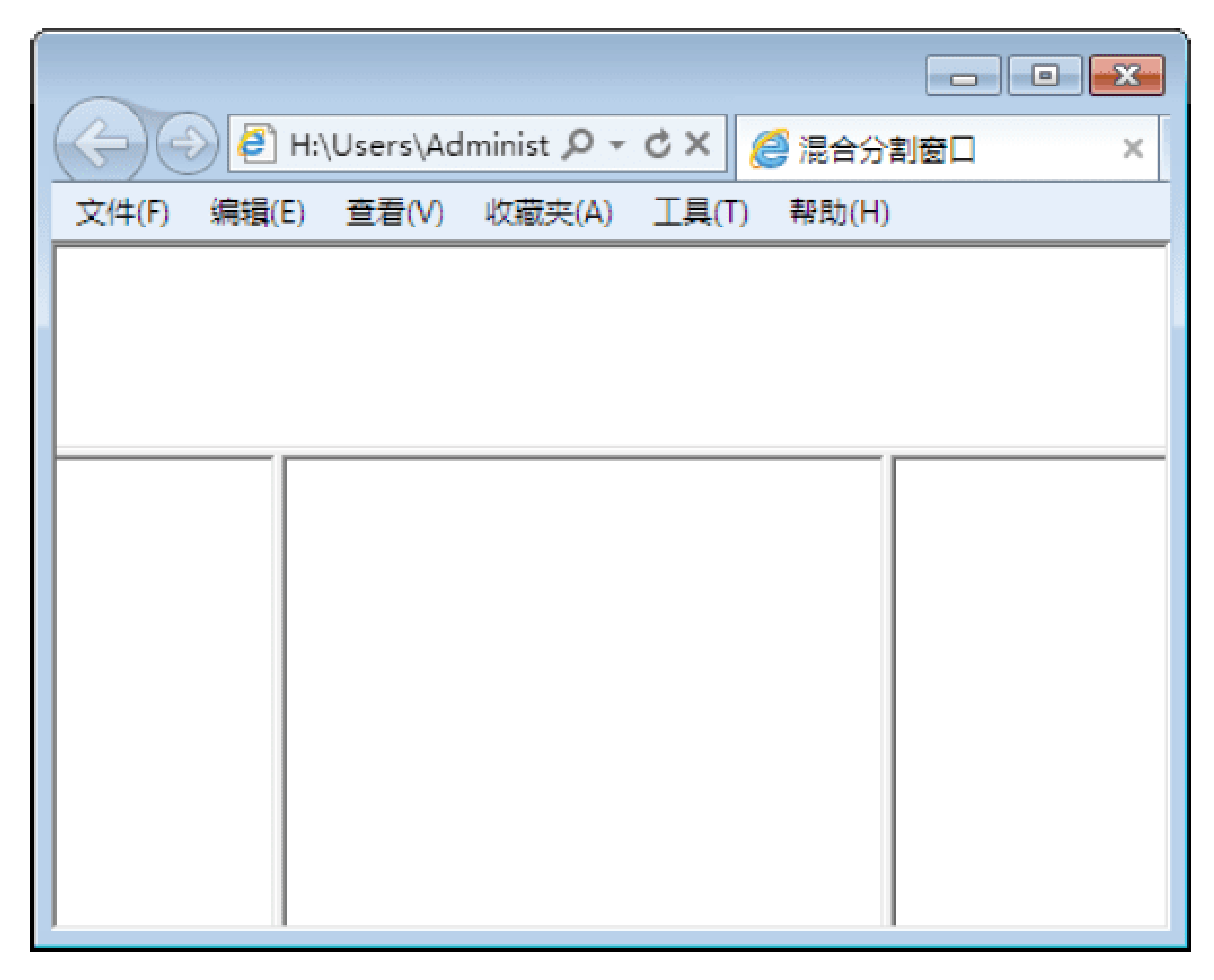


图 8-3　混合分割窗口

8.2.4　设置框架边框

在默认情况下，框架窗口的四周有一条边框线，通过 frameborder 属性可以调整边框线的显示情况。语法结构如下。

```
<frame frameborder="value">
```



frameborder 的取值只能为 0 或 1。如果 value 取值为 0，那么边框线将会被隐藏；如果 value 取值为 1，边框线将会显示。在 frameset 中设置将会对整个框架有效，在 frame 中设置则只对当前框架有效。

【例 8.4】（实例文件：ch08\8.4.html）

```
<html>
<head>
<title>设置框架边框</title>
</head>
<frameset rows="20%,55%,25%">
<frame frameborder="1">
   <frameset cols="35%,65%" frameborder="0">
<frame>
<frame>
   </frameset>
<frame frameborder="0">
</frameset>
</html>
```

运行这段代码，可以看到页面中的部分边框被隐藏，如图 8-4 所示。当鼠标移动到窗口中

间时会变成⇔，按下鼠标就会看到隐藏的边框，如图 8-5 所示。

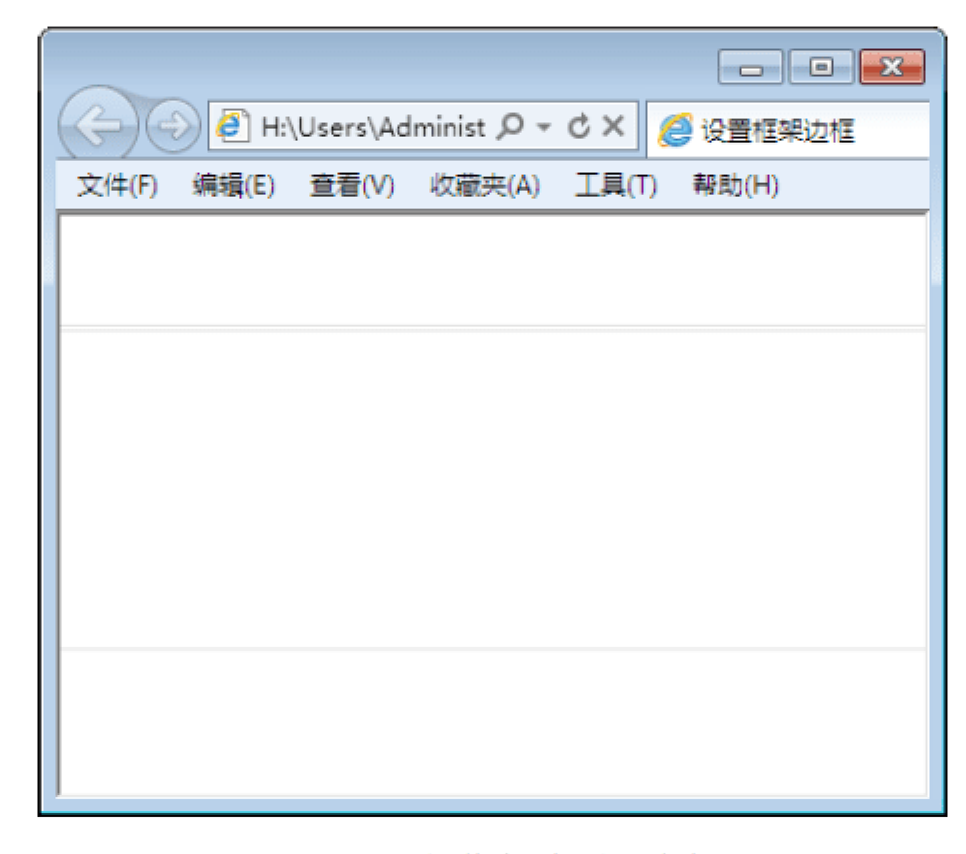


图 8-4　隐藏框架的边框

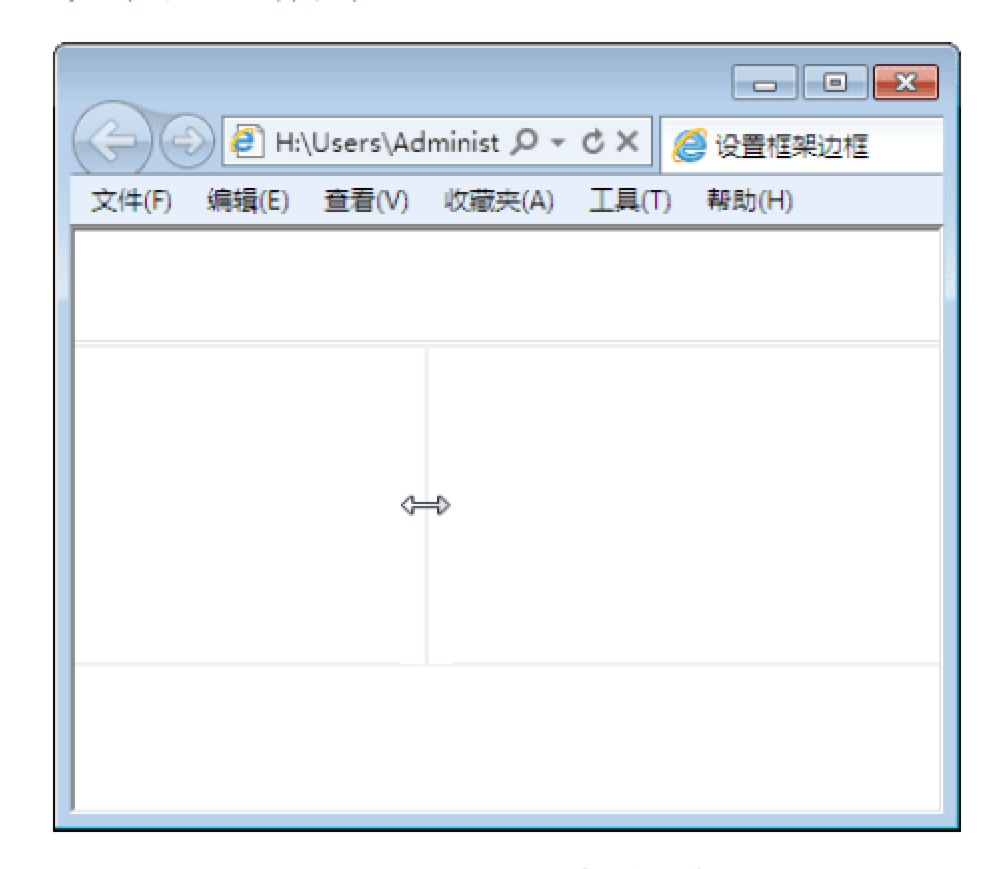


图 8-5　按下鼠标的效果

8.2.5　设置框架的边框宽度

框架的边框宽度在默认情况下是 1 个像素，通过参数 framespacing 可以调整其大小。其语法结构如下：

```
<frameset framespacing="边框宽度">
```

边框宽度就是在页面中各个边框之间的线条宽度，以像素为单位。这一参数只能对框架集使用，对单个框架无效。

【例 8.5】（实例文件：ch08\8.5.html）

```
<html>
<head>
<title>设置框架的边框宽度</title>
</head>
<frameset rows="30%,70%" framespacing="10">
<frame>
  <frameset cols="20%,55%,25%" framespacing="20">
<frame>
<frame>
<frame>
  </frameset>
</frameset>
</html>
```

运行这段程序之后的显示效果如图 8-6 所示。

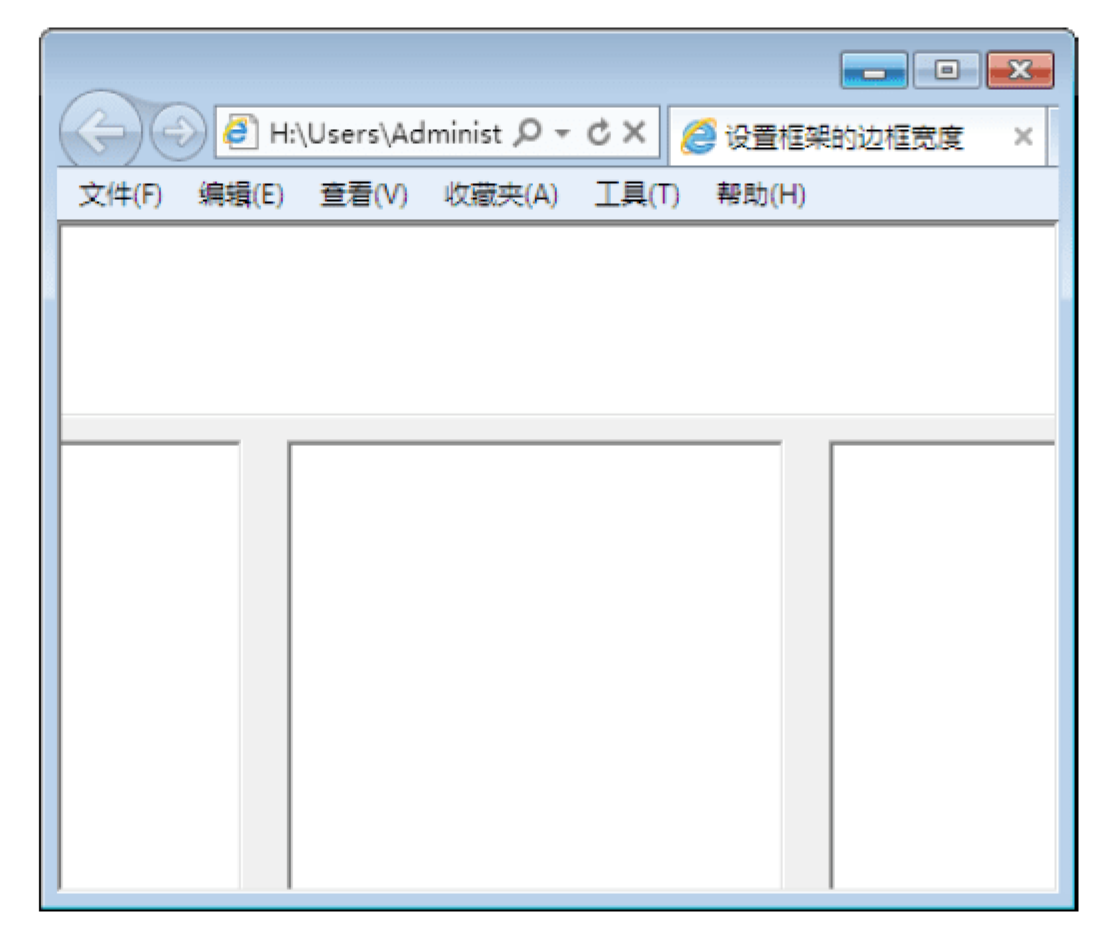


图 8-6　设置框架的边框宽度

8.2.6　设置框架的边框颜色

使用参数 bordercolor 可以设置框架集的边框颜色。其语法结构如下。

```
<frameset bordercolor="颜色代码">
```



该参数只对整个框架集有效，对于单个框架无效。

【例 8.6】（实例文件：ch08\8.6.html）

```
<html>
<head>
<title>设置框架的边框颜色</title>
</head>
<frameset rows="30%,70%" framespacing="10" bordercolor ="green">
<frame>
    <frameset cols="20%,55%,25%" framespacing="20" bordercolor ="red">
<frame>
<frame>
<frame>
  </frameset>
</frameset>
</html>
```

运行这段代码后，可以看到两个框架集设置了不同的边框颜色，效果如图 8-7 所示。

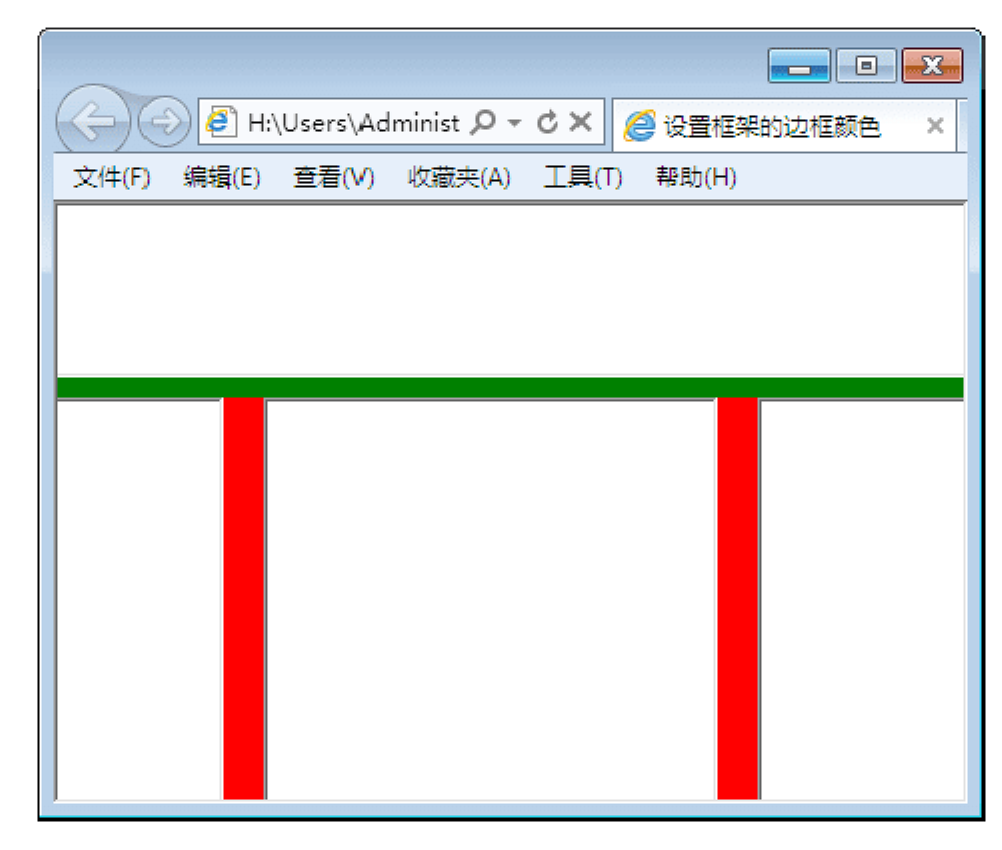


图 8-7 设置边框颜色

8.3 设置窗口属性

在框架结构的页面中，每一个框架窗口都有一个显示页面，这个页面称为框架页面。框架页面的属性设置都在<frame>标记里进行。

8.3.1 页面源文件

页面的源文件包括多种类型，如图片、网页文件、动画等。框架结构中的各个页面都是一个单独的文件，而这些文件是通过 src 参数进行设置的。其语法结构如下。

```
<frame src="页面源文件地址">
```



页面文件是框架页面的具体内容所在，对于没有设置源文件的框架，即空白页面，是没有任何作用的。页面的源文件可以是正常的 HTML 文件，也可以是一个图片或者其他类型的文件。

为了说明框架的参数效果，下面设置一个框架页面的内容。

【例 8.7】（实例文件：ch08\8.7.html）

```
<html>
<head>
<title>诗歌</title>
</head>
<body>
《沁园春  雪》
<p>北国风光，千里冰封，万里雪飘。</p>
<p>望长城内外，惟余莽莽；大河上下，顿失滔滔。</p>
<p>山舞银蛇，原驰蜡象，欲与天公试比高。</p>
<p>须晴日，看红装素裹，分外妖娆。</p>
```

```
<p>江山如此多娇，引无数英雄竞折腰。</p>
<p>惜秦皇汉武，略输文采；唐宗宋祖，稍逊风骚。</p>
<p>一代天骄，成吉思汗，只识弯弓射大雕。俱往矣，数风流人物，还看今朝。</p>
</body>
</html>
```

将这一文件命名为 8.7.html，将其保存在框架集文件的同一目录下。下面就可以设置页面源文件了。

【例 8.8】（实例文件：ch08\8.8.html）

```
<html>
<head>
<title>设置页面源文件</title>
</head>
<frameset rows="40%,60%">
<frame src="图 01.jpg">
<frame src="8.7.html">
</frameset>
</html>
```

在 IE 9.0 中浏览效果如图 8-8 所示。



图 8-8 设置页面源文件

8.3.2 设置页面名称

页面名称是为了便于页面的查找和链接所提供的一个属性。框架的页面名称中不允许包含特殊字符、连字符、空格等，必须是单个的单词或者字母组合。其语法结构如下：

```
<frame src="页面文件地址" name="页面名称">
```

【例 8.9】（实例文件：ch08\8.9.html）

```
<html>
```

```
<head>
<title>设置页面名称</title>
</head>
<frameset rows="40%,60%">
<frame src="图 01.jpg" name="pic">
<frame src="8.7.html" name="text">
</frameset>
</html>
```

这段代码分别为上下两个框架页面命名为 pic 和 text，运行代码时并不显示框架名，因此这样做不会影响框架的显示效果。在 IE 9.0 中浏览效果如图 8-9 所示。



图 8-9　设置页面名称

8.3.3　固定窗口的尺寸

在具有框架的页面中，当鼠标拖动框架的边框时，框架窗口的尺寸也会随之变化。如果不希望窗口的大小发生变化，可以用 noresize 参数来实现。其语法结构如下。

```
<frame src="页面文件地址" noresize>
```



在 frame 标记中添加 noresize 参数，就表示框架窗口的尺寸不能改变。

【例 8.10】（实例文件：ch08\8.10.html）

```
<html>
<head>
<title>固定窗口的尺寸</title>
</head>
<frameset rows="30%,70%">
<frame src="图 01.jpg" name="pic" noresize>
<frame src="8.7.html" name="text">
</frameset>
```

```
</html>
```

在 IE 9.0 中浏览效果如图 8-10 所示。将鼠标放置在框架边框上时，光标不会变为双箭头形状，也无法拖动框架的边框。

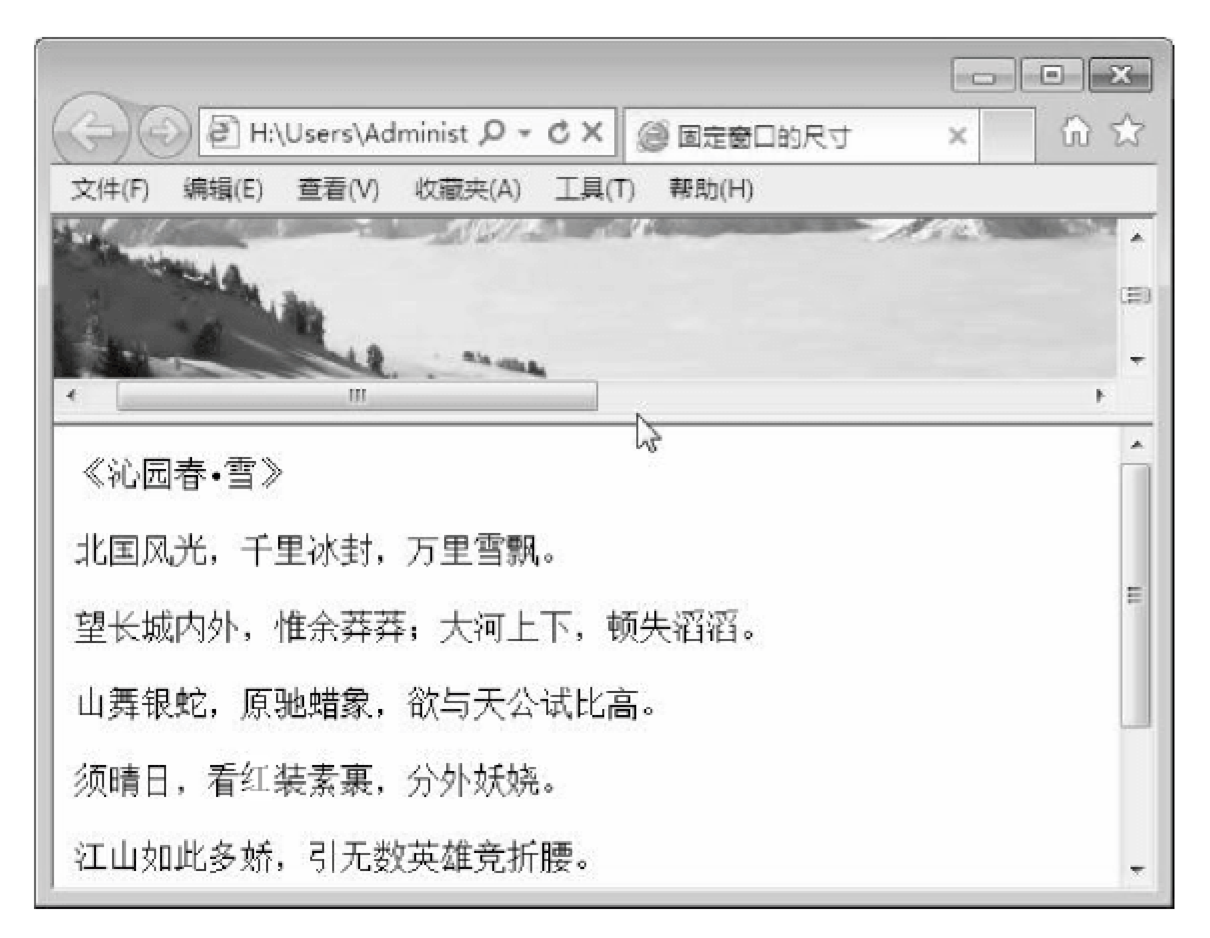


图 8-10　固定窗口的尺寸

8.3.4　边框与页面内容的水平边距

框架页面与 HTML 中的表格一样，也可以设置边框与页面内容的水平边距。其语法结构如下。

```
<frame src="页面文件地址" marginwidth="水平边距">
```



水平边距用于设置页面的左右边缘与框架边框的距离，单位是像素。

首先创建一个源文件 8.11.html。

【例 8.11】（实例文件：ch08\8.11.html）

```
<html>
<head>
<title>设置水平边距</title>
</head>
<body>
<br>
<h3><font color="#009933">春天里的秋天</font></h3>
<hr>
<font color="#009966"><br>
春天，枯黄的原野变绿了。新绿的叶子在枯枝上长出来。阳光温柔地对着每个人微笑，鸟儿在歌唱飞翔。花开放着，红的花，白的花，紫的花。星闪耀着，红的星，绿的星，白的星。蔚蓝的天，自由的风，梦一般美丽的爱情。 每个人都有春天。无论是你，或者是我，每个人在春天里都可以有欢笑，有爱情，有陶醉。 然
```

```
而秋天在春天里哭泣了。这一个春天，在迷人的南国的古城里，我送走了我的一段光阴。 秋天的雨落了，但是又给春天的风扫尽了。在雨后的一个晴天里，我同两个朋友走过泥泞的道路。走过石板的桥，走过田畔的小径，去访问一个南国的女性，一个我不曾会过面的疯狂的女郎。
    </font>
    </body>
    </html>
```

在 IE 9.0 中浏览效果如图 8-11 所示。

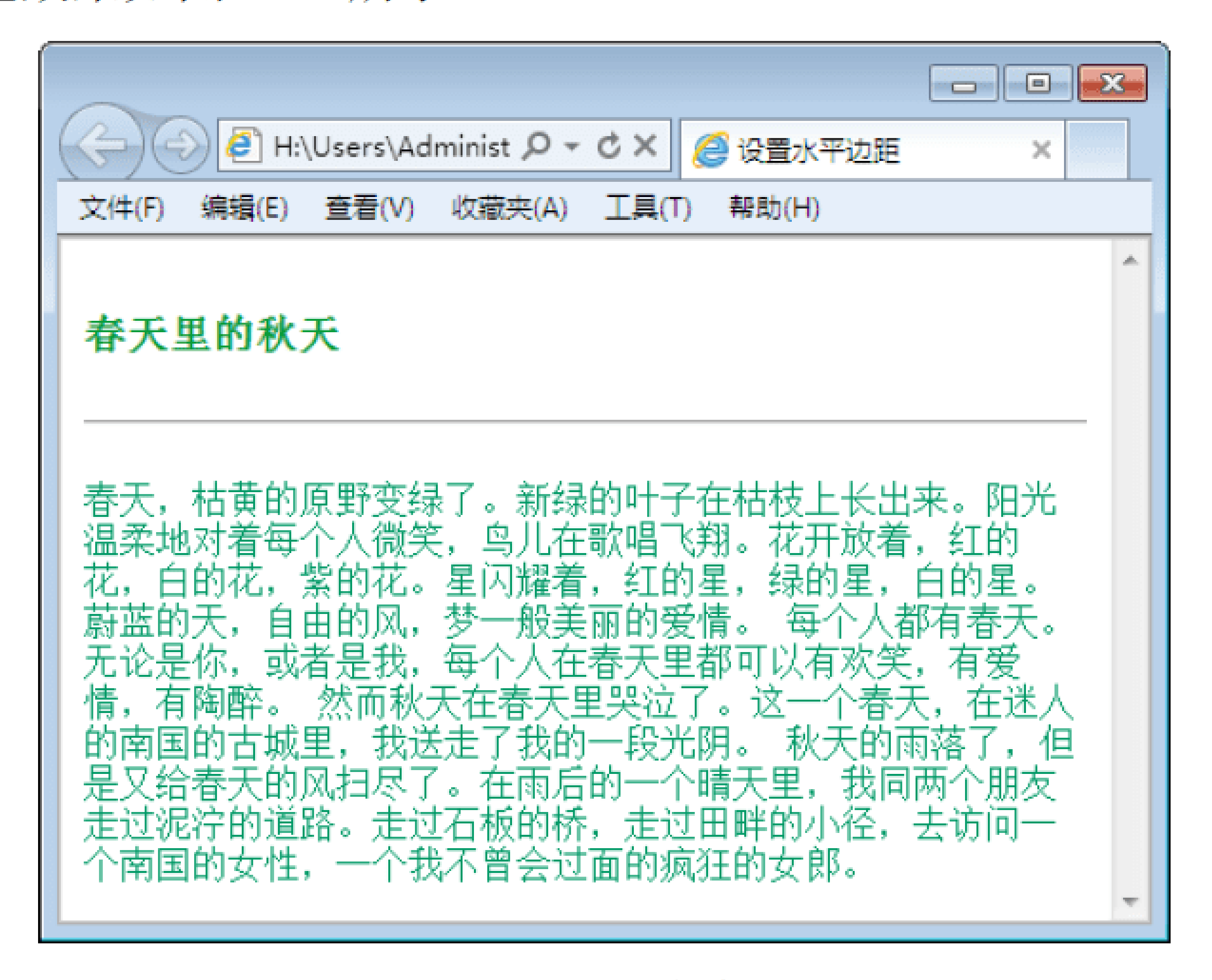


图 8-11　预览效果

下面就可以创建框架集代码了。

【例 8.12】（实例文件：ch08\8.12.html）

```
<html>
<head>
<title>设置水平边距</title>
</head>
<frameset cols="35%,65%">
<frame src="图 02.jpg" name="pic" >
<frame src="8.11.html" name="text" marginwidth="70">
</frameset>
</html>
```

在 IE 9.0 中浏览效果如图 8-12 所示，可以看到右侧的框架页面中，文本内容与框架的边框之间设置了很大的空间，从而显得宽松很多。



图 8-12　设置水平边距的预览效果

8.3.5　边框与页面内容的垂直边距

通过 marginheight 参数可以设置边框与页面的垂直边距。其语法结构如下：

```
<frame src="页面文件地址" marginheight="垂直边距">
```



垂直边距用来设置页面的上下边缘与框架边框的距离，单位是像素。

【例 8.13】（实例文件：ch08\8.13.html）

```
<html>
<head>
<title>设置垂直边距</title>
</head>
<frameset cols="35%,65%">
<frame src="图 02.jpg" name="pic" >
<frame src="8.11.html" name="text" marginwidth="70" marginheight="90">
</frameset>
</html>
```

在 IE 9.0 中浏览效果如图 8-13 所示，可以看到右侧的框架页面中，文本内容与框架的上下边框之间设置了很大的空间。

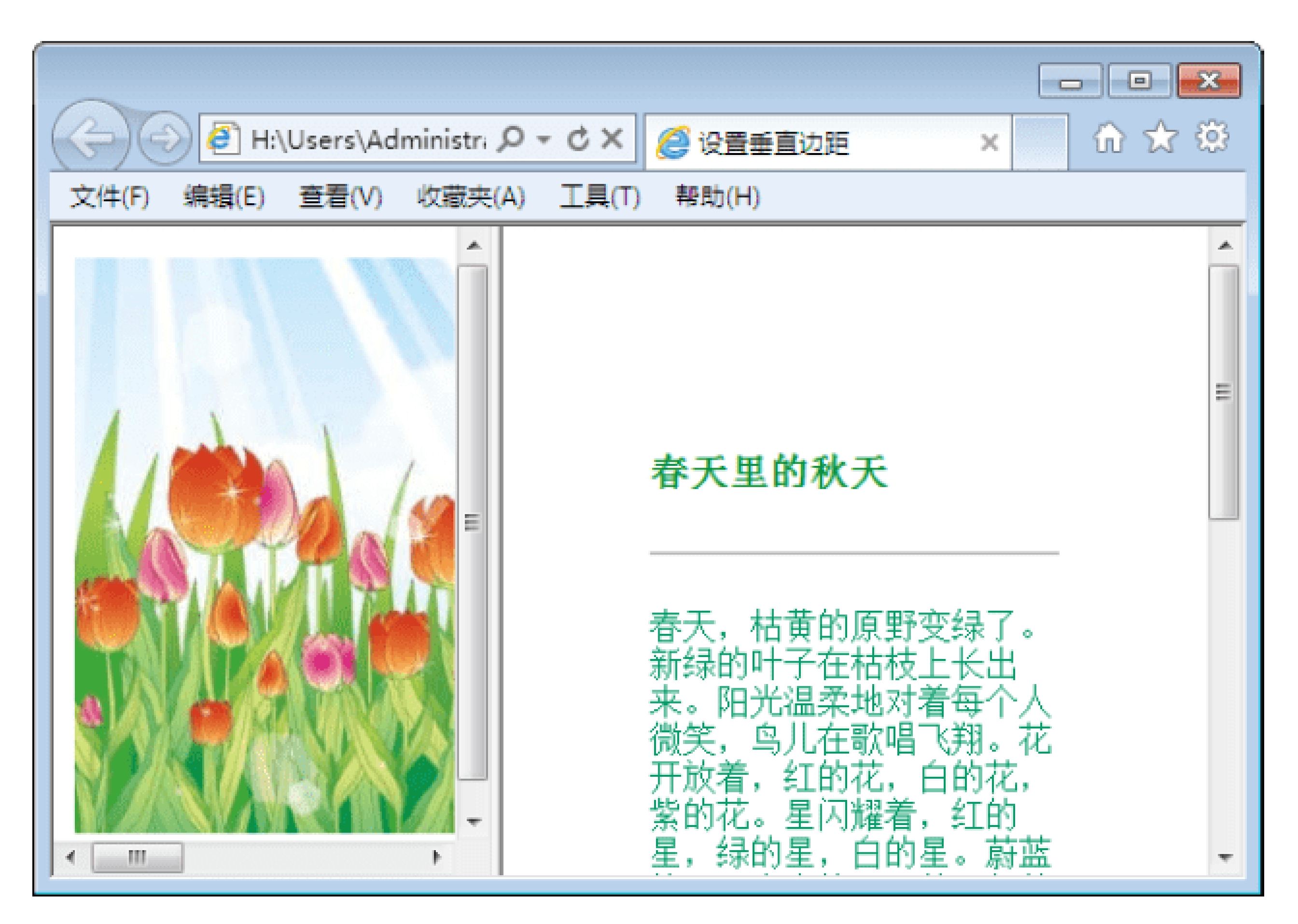


图 8-13　设置垂直边距的效果

8.3.6　设置框架滚动条显示

默认情况下，当页面的内容超出浏览器窗口的大小时，页面会自动添加滚动条以方便用户浏览。但如果用户不希望显示滚动条，可以通过设置 scrolling 参数来实现。语法结构如下。

```
<frame src="页面源文件" scrolling="是否显示滚动条">
```



scrolling 参数在这里只能取 Yes、No 或 Auto 三个值中的一个。其中，Yes 表示一直显示滚动条，而 No 则表示无论什么情况都不显示滚动条；Auto 是系统的默认值，它是根据具体内容来调整的，也就是说当页面长度超出浏览器窗口的范围时就会自动显示滚动条。

【例 8.14】（实例文件：ch08\8.14.html）

```
<html>
<head>
<title>设置垂直边距</title>
</head>
<frameset cols="35%,65%">
<frame src="图 02.jpg" name="pic" scrolling="Yes">
<frame src="8.11.html" name="text" marginwidth="70" marginheight="90" scrolling="No">
</frameset>
</html>
```

在 IE 9.0 中浏览效果如图 8-14 所示，与图 8-13 相对比，可以看出设置 scrolling 为 Yes 时，即使页面内容远远小于浏览器的大小，也会显示滚动条；而当 scrolling 设置为 No 时，即使内容超出了浏览器的范围，也没有显示滚动条。

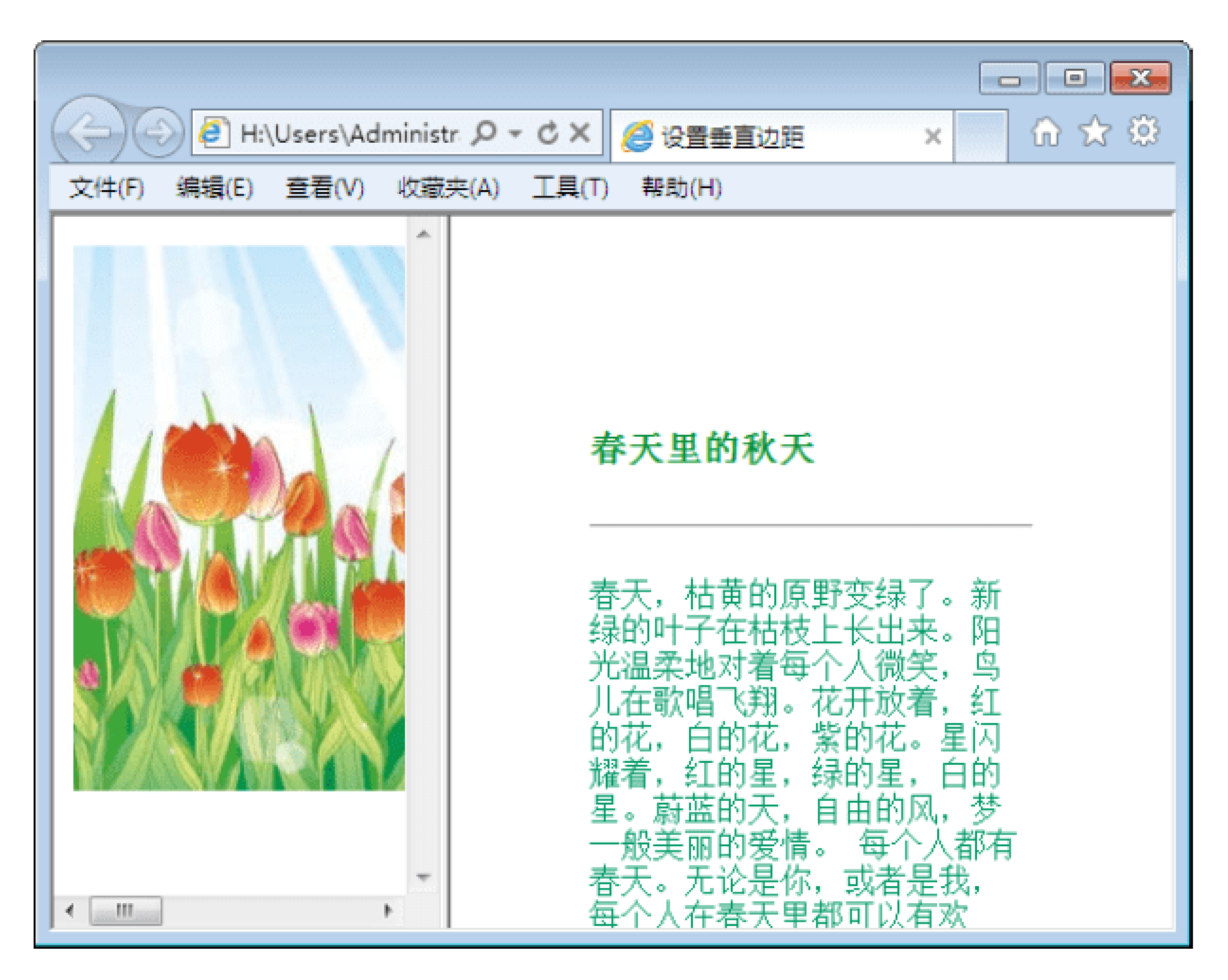
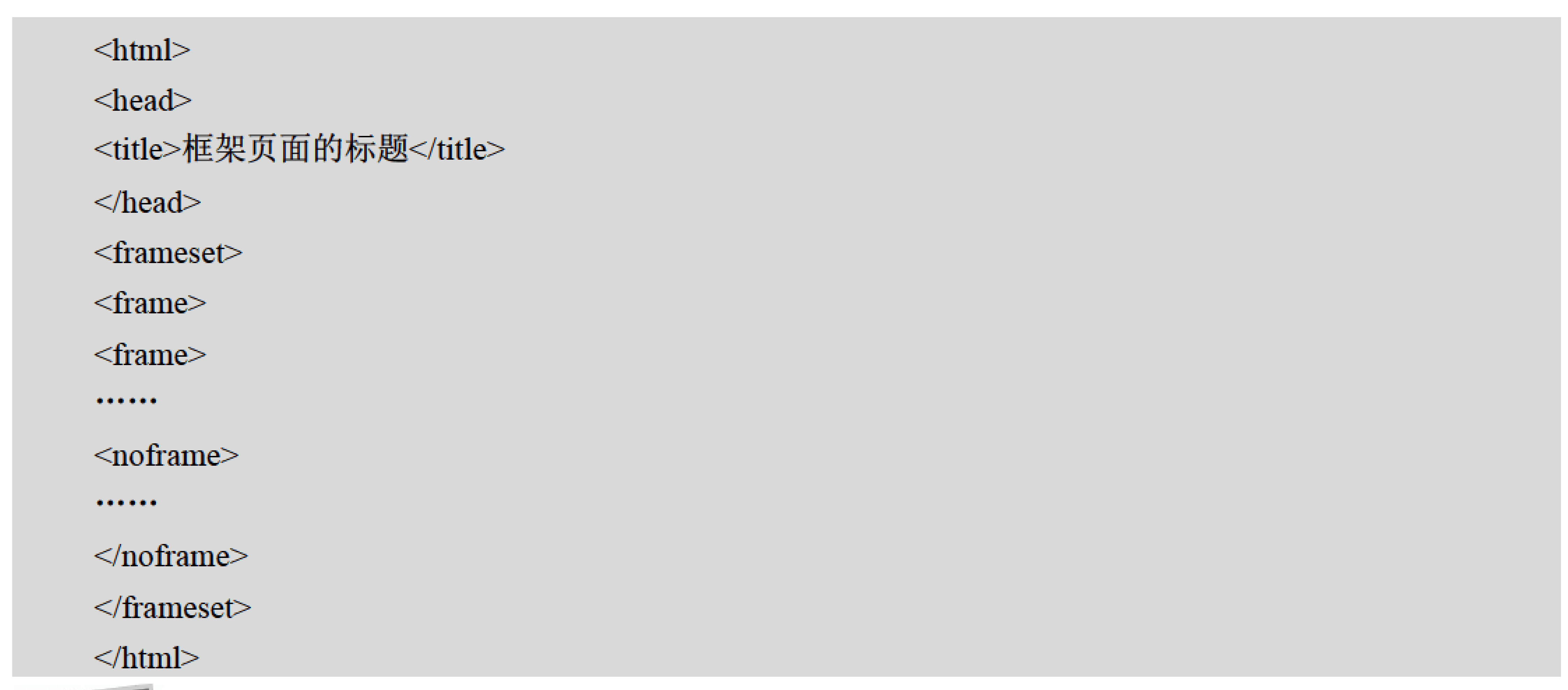


图 8-14　设置滚动条的效果

8.3.7　不支持框架标签

对于一些低版本的浏览器来说，如果不支持框架结构就无法打开框架页面，但这样还不够，因为它并不清楚为什么打不开页面。通过 noframe 参数可以设置一些内容（包括文字或图片）来告诉浏览者其浏览器无法打开框架页面。语法结构如下。



```
<html>
<head>
<title>框架页面的标题</title>
</head>
<frameset>
<frame>
<frame>
……
<noframe>
……
</noframe>
</frameset>
</html>
```



在该语法中，<noframe>和</noframe>标记之间的部分就是在不支持框架的浏览器中所要显示的内容。可以与<body>标记一样在其中添加内容。

【例 8.15】（实例文件：ch08\8.15.html）

```
<html>
<head>
```

```
<title>不支持框架页面</title>
</head>
<frameset cols="35%,65%">
<frame src="图 02.jpg" name="pic" >
<frame src="8.11.html" name="text">
<noframe>
<center>
    <p>对不起，您的浏览器不支持框架页面，因此无法显示此页面！</p>
<img src="图 03.jpg">
</center>
</noframe>
</frameset>
```

在 IE 9.0 中浏览效果如图 8-15 所示，当用户的浏览器不支持框架结构时，就直接显示<noframe>标记内的页面内容。



图 8-15　显示<noframe>标记内的内容

8.4　内联框架

内联框架是一种较为特殊的框架，它是在浏览器窗口中嵌套子窗口，也就是整个页面并不是框架页面，但是却包含一个框架窗口，在框架窗口内显示相应的页面内容。语法结构如下：

```
<iframe "页面源文件">
</iframe>
```

8.4.1　内联框架的主要参数

与普通框架结构类似，内联框架也可以设置很多参数，如表 8-1 所示。

表 8-1　内联框架的参数设置

参数	含义
src	内联框架页面的源文件地址
width	内联框架在页面中显示的宽度
height	内联框架在页面中显示的高度

（续表）

参数	含义
Align	内联框架页面在浏览器中的对齐方式，可以为左对齐、右对齐或居中对齐
name	设定框架页面的名称
marginwidth	设置页面与边框的水平间距
marginheight	设置页面与边框的垂直间距

从表中可以看出，很多普通框架页面的属性在内联框架页面中同样适用，如 name 等，而在普通框架中只对框架集有效的一些参数在这里同样可以设置，如 frameborder 等，此外内联框架页面还可以设置大小和对齐方式。而对于内联框架来说，框架边框的宽度和颜色是无法设置的，即与普通框架相比，内联框架中不包含 framespacing 和 bordercolor 参数。

本节将主要通过设置框架页面的参数和不设置进行对比，从而清晰地说明各个参数的功能。为了达到这一目的，需要进行如下的准备工作：创建两个普通的 HTML 页面，其中一个作为内联框架的载体，命名为 float.html；另一个作为内联框架页面的源文件，命名为 source.html。float.html 文件的源代码如下：

【例 8.16】（实例文件：ch08\float.html）

```
<html>
<head>
<title>未添加内联框架的页面</title>
</head>
<body>
<font size="5" color="#CC0000">在这一个页面中将会添加一个内联框架页面</font>
<hr size=2>
</body>
</html>
```

在 IE 9.0 中浏览效果如图 8-16 所示。

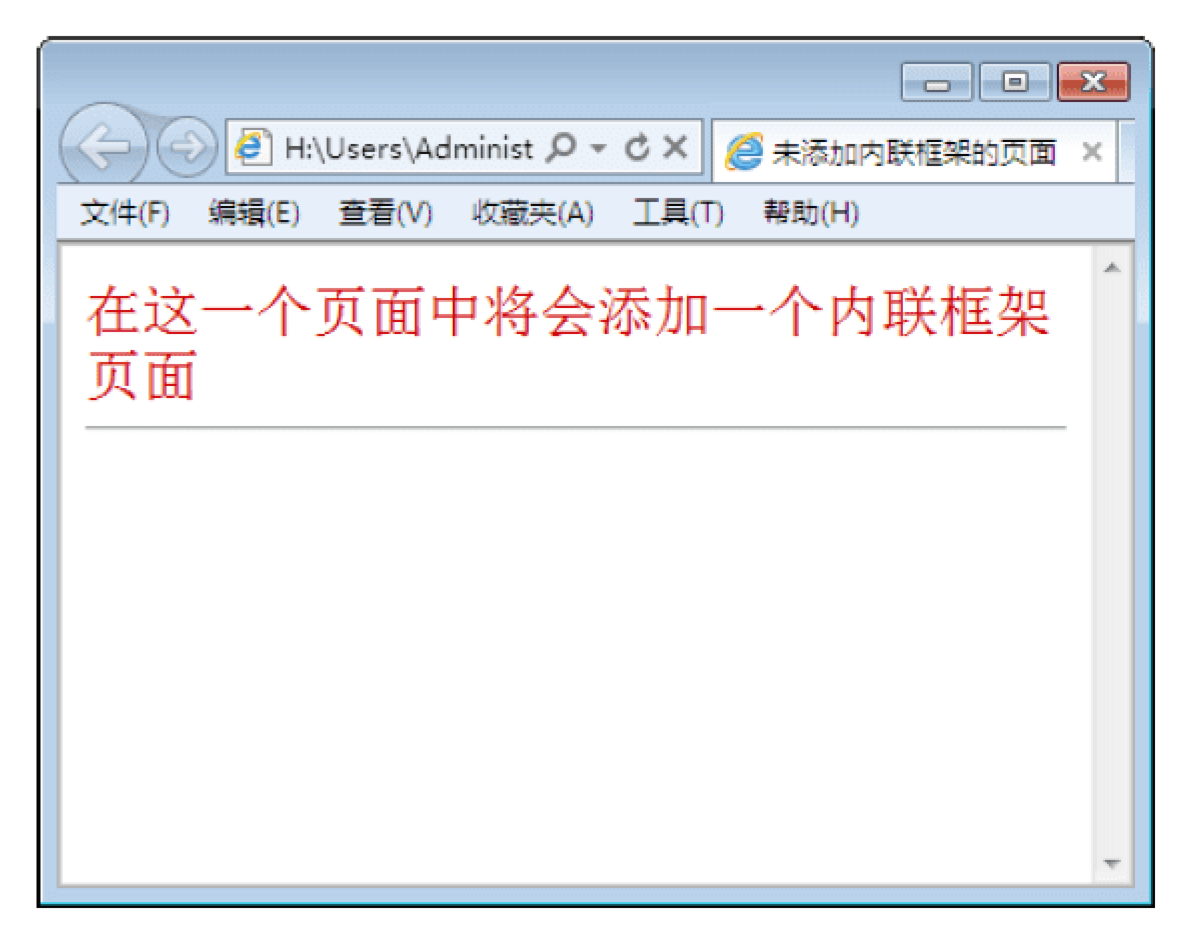


图 8-16　未添加内联框架的页面

Source.html 文件的源代码如下。

【例 8.17】（实例文件：ch08\source.html）

```
<html>
<head>
<title>内联框架的页面内容</title>
</head>
<body>
<font size="5" color="#000099">泰坦尼克号</font><br><br>
<image src="图 04.jpg" align="left">
《泰坦尼克号》作为电影史上三大凄美不朽爱情影片之一，是一部荡气回肠的爱情经典之作，具有甚高的催泪效果。<br>
《泰坦尼克号》是一部风靡全球的美国爱情故事片，也是西方电影在东方获得成功的经典。<br>
<hr>
剧情介绍：<br>
该影片向众人讲述了一段动人的爱情故事：在泰坦尼克号上的罗丝因不想嫁给未婚夫卡尔，准备跳海自杀，被幸运赢得船票的杰克所救，两人由此认识。在随后的几天里，杰克带她参加三等舱的舞会，给她画素描（布洛克打捞起的那张），教她吐痰……他们两人很快坠入了情网。但是这份甜美的爱情还没过多久，泰坦尼克号便撞上冰山了。杰克为了罗丝能够存活下来，不断地鼓励她，而自己却冻死在寒冷的海水里。罗丝得救了，但她从没有忘记杰克，不仅仅是因为他们有一段凄美的爱情，还因为杰克带给了她欢乐，也给了她希望和未来。
</body>
</html>
```

在 IE 9.0 中浏览效果如图 8-17 所示。



图 8-17　内联框架的页面内容

完成了两个页面文件之后，下面就可以开始设置内联框架页面的内容了。

8.4.2　必需参数：页面源文件

内联框架中最基本的参数就是 src，它用来设置内联框架页面的源文件地址，也是内联框架页面必需的参数。语法结构如下：

```
<iframe src="页面源文件">
```

下面将文件 source.html 作为内联框架页面的源文件插入到 HTML 文件 float.html 中。

【例 8.18】（实例文件：ch08\8.18.html）

```
<html>
<head>
<title>页面源文件</title>
</head>
<body>
<font size="5" color="#CC0000">在这一个页面中将会添加一个内联框架页面</font>
<hr size=2>
<iframe src="source.html">
</iframe>
</body>
</html>
```

在 IE 9.0 中浏览效果如图 8-18 所示。



图 8-18　页面源文件

8.4.3　特有属性：框架大小

内联框架特有的参数主要包括框架的页面大小以及对齐方式。在普通框架结构中，由于框架就是整个浏览器窗口，因此不需要设置其大小。但是在浮动框架中，框架是插入到普通 HTML 页面中的，所以可以调整整个框架的大小。语法结构如下：

```
<iframe src="浮动框架页面源文件" width="页面宽度" height="页面高度">
```



在该语法中，页面的宽度和高度值都以像素为单位。

下面在 float.html 文件的基础上设置浮动框架页面的大小。

【例 8.19】（实例文件：ch08\8.19.html）

```
<html>
<head>
<title>设置框架页面的大小</title>
</head>
<body>
<font size="5" color="#CC0000">在这一个页面中将会添加一个浮动框架页面</font>
<hr size=2>
<iframe src="source.html" width="550" height="310">
</iframe>
</body>
</html>
```

在 IE 9.0 中浏览效果如图 8-19 所示。与图 8-18 相对比，可以看到页面中的框架大小被调整得更加协调。



图 8-19　设置框架页面的大小

8.4.4　特有属性：对齐方式

除了设置框架大小的参数之外，设置框架的对齐方式也是浮动框架页面所特有的参数。其语法结构如下。

```
<iframe src="浮动框架页面源文件" align="对齐方式">
```



在该语法中，对齐方式用来设置浮动框架页面相对于浏览器窗口的水平位置。它可以取的值包括左对齐（left）、右对齐（right）和居中对齐（center）。

下面在 float.html 文件的基础上设置浮动框架页面的对齐方式。

【例 8.20】（实例文件：ch08\8.20.html）

```
<html>
<head>
<title>设置对齐方式</title>
</head>
<body>
<font size="5" color="#CC0000">在这一个页面中将会添加一个浮动框架页面</font>
<hr size=2>
<iframe src="source.html" width="550" height="310" align="center">
</iframe>
</body>
</html>
```

在 IE 9.0 中浏览效果如图 8-20 所示。与图 8-19 相对比，可以看到页面中的框架位置被调整了，由原来的左对齐变成了居中对齐。



图 8-20　设置框架页面的对齐方式

8.4.5　共有参数设置

普通框架页面和内联框架页面所共有的一些参数称为内联框架的基本属性，主要包括 src、name、marginwidth、marginheight 和 scrolling。由于这些参数的设置方法与普通框架类似，这里不再赘述。具体的语法结构如下。

```
<iframe 浮动框架页面源文件" name="页面名称" marginwidth="边框与页面的水平距离"
marginheight="边框与页面的垂直距离" scrolling="是否显示滚动条">
</iframe>
```



在这里，这些参数的取值范围与普通框架相同，因此不再重复，只通过下面的实例说明这些属性的设置效果。

【例 8.21】（实例文件：ch08\8.21.html）

```
<html>
<head>
<title>设置框架的共有参数</title>
</head>
<body>
<font size="5" color="#CC0000">在这一个页面中将会添加一个浮动框架页面</font>
<hr size=2>
<iframe src="source.html" width="550" height="310" align="center" name="public" marginwidth="60" marginheight="40" scrolling="No">
</iframe>
</body>
</html>
```

在 IE 9.0 中浏览效果如图 8-21 所示。对比图 8-20，可以看到框架页面的内容与边框增加了一定的空间，而且在页面中虽然无法显示全部的内容，但是并没有出现滚动条。



图 8-21 设置框架页面的共同参数

8.4.6 其他参数

在内联框架页面中，还有另外一个特殊的参数可以设置，即框架边框显示属性

frameborder。在普通框架中，该参数既对整个框架集有效，也对单个框架有效。同样地，在内联框架中也可以设置这一参数。语法结构如下：

```
<iframe 浮动框架页面源文件" frameborder="是否显示">
</iframe>
```



在这里，这些参数的取值范围与普通框架相同。其中 src 是浮动框架的页面源文件；frameborder 只能取 0 或 1，0 表示不显示边框，1 为默认取值，表示显示边框。

【例 8.22】（实例文件：ch08\8.22.html）

```
<html>
<head>
<title>设置框架页面的边框</title>
</head>
<body>
<font size="5" color="#CC0000">在这一个页面中将会添加一个浮动框架页面</font>
<hr size=2>
<iframe src="source.html" width="550" height="310" align="center" name="public" marginwidth="60"
marginheight="40" scrolling="No" frameborder="0">
</iframe>
</body>
</html>
```

在 IE 9.0 中浏览效果如图 8-22 所示。对比图 8-20，可以看到浮动框架的页面边框不见了。



图 8-22 设置框架页面的边框

8.5　框架与链接

大部分情况下，在页面中添加框架是为了更好地对页面内容进行导航，因此常常是通过框架来链接不同的内容。对于框架页面来说，设置页面的链接需要使用 target 参数，这一参数的取值不同于它在普通 HTML 页面中的取值。

8.5.1　设置普通框架结构的链接

普通的框架结构之间通过 target 参数互相链接起来。一般情况下，一个页面中会有一个框架窗口作为导航页面，里面添加了对另外一个框架的内容的链接设置，而这些链接则是通过 target 实现的。

下面通过一个具体的实例来讲解关于框架之间的链接方法。首先需要设定一个框架集文件，用于确定框架页面的整体布局，这里将文件命名为“民间传说.html”。

【例 8.23】（实例文件：ch08\民间传说.html）

```
<html>
<head>
<title>框架页面的整体结构</title>
</head>
<frameset rows="200,*">
<frame src="navig.html" name="navig" >
<frame src="content-1.html" name="content">
</frameset>
</html>
```

此处按照常见网站的上下结构进行布局，将页面上面的框架窗口设置为 200 像素高，下面的窗口则分割剩下的部分。在该文件中，设置了上面的框架窗口页面为导航页面，命名为 navig，同时设置其默认打开的页面名为 navig.html。同时还设置该页面窗口不显示滚动条，而且不允许调整窗口大小。在该文件中还设置了下面的框架窗口为内容页面，命名为 content，设置默认打开的页面名为 content-1.html。

下面就介绍这两个文件的具体编码。其中，navig.html 文件将放置在框架的上方，作为页面的导航页。

【例 8.24】（实例文件：ch08\ navig.html）

```
<html>
<head>
<title>导航页面</title>
</head>
<body>
<center>
  <img src="图 05.jpg"><br><br>
  <a href="content-1.html" target="content">白蛇传</a>
```

```
                       
    <a href="content-2.html" target="content">神笔马良</a>
                       
    <a href="content-3.html" target="content">菊花仙子</a>
</center>
</body>
</html>
```

在这个文件代码中，target 的值是设置将要显示页面内容的框架名称，此处设置为 content，说明设置的这 3 个内容页面将在 content 框架窗口中显示。

content-1.html 文件是默认的内容页面，其代码如下。

【例 8.25】（实例文件：ch08\content-1.html）

```
<html>
<head>
<title>内容页面</title>
</head>
<body>
<center>
    <font size=5>白蛇传</font><br><br>
</center>
     清明时分，西湖岸边花红柳绿，断桥上面游人如梭，真是一幅春光明媚的美丽画面。突然，从西湖底悄悄升上来两个如花似玉的姑娘，怎么回事？人怎么会从水里出来呢？原来，她们是两条修炼成了人形的蛇精，虽然如此，但她们并无害人之心，只因羡慕世间的多彩人生，才一个化名叫白素贞，一个化名叫小青，来到西湖边游玩。<br>
     偏偏老天爷忽然发起脾气来，霎时间下起了倾盆大雨，白素贞和小青被淋得无处藏身，正发愁呢，突然只觉头顶多了一把伞，转身一看，只见一位温文尔雅、白净秀气的年轻书生撑着伞在为她们遮雨。白素贞和这小书生四目相交，都不约而同地红了红脸，相互产生了爱慕之情。小青看在眼里，忙说：“多谢！请问客官尊姓大名。”那小书生道：“我叫许仙，就住在这断桥边。”白素贞和小青也赶忙做了自我介绍。从此，他们三人常常见面，白素贞和许仙的感情越来越好，过了不久，他们就结为夫妻，并开了一间“保和堂”药店，小日子过得可美了！<br>
     由于“保和堂”治好了很多很多疑难病症，而且给穷人看病配药还分文不收，所以药店的生意越来越红火，远近来找白素贞治病的人越来越多，人们将白素贞亲切地称为白娘子。可是，“保和堂”的兴隆、许仙和白娘子的幸福生活却惹恼了一个人，谁呢？那就是金山寺的法海和尚。因为人们的病都被白娘子治好了，到金山寺烧香求菩萨的人就少多了，香火不旺，法海和尚自然就高兴不起来了。这天，他又来到“保和堂”前，看到白娘子正在给人治病，不禁心内妒火中烧，再定睛一瞧，哎呀！原来这白娘子不是凡人，而是条白蛇变的！<br>
     法海虽有点小法术，但他的心术却不正。看出了白娘子的身份后，他就想拆散许仙白娘子夫妇、搞垮“保和堂”。于是，他偷偷把许仙叫到寺中，对他说：“你娘子是蛇精变的，你快点和她分手吧，不然，她会吃掉你的！”许仙一听，非常气愤，他想：我娘子心地善良，对我的情意比海还深。就算她是蛇精，也不会害我，何况她如今已有了身孕，我怎能离弃她呢！法海见许仙不上他的当，恼羞成怒，便把许仙关在了寺里。<br>
```

```
         在“保和堂”里，白娘子正焦急地等待许仙回来。一天、两天，左等、右等，白娘子心急如焚。终于打听到原来许仙被金山寺的法海和尚给“留”住了，白娘子赶紧带着小青来到金山寺，苦苦哀求，请法海放回许仙。法海见了白娘子，一阵冷笑，说道：“大胆妖蛇，我劝你还是快点离开人间，否则别怪我不客气了！”白娘子见法海拒不放人，无奈，只得拔下头上的金钗，迎风一摇，掀起滔滔大浪，向金山寺直逼过去。法海眼见水漫金山寺，连忙脱下袈裟，变成一道长堤，拦在寺门外。大水涨一尺，长堤就高一尺，大水涨一丈，长堤就高一丈，任凭波浪再大，也漫不过去。再加上白娘子有孕在身，实在斗不过法海，后来，法海使出欺诈的手法，将白娘子收进金钵，压在了雷峰塔下，把许仙和白娘子这对恩爱夫妻活生生地拆散了。<br>
         小青逃离金山寺后，数十载深山练功，最终打败了法海，将他逼进了螃蟹腹中，救出了白娘子，从此，她和许仙以及他们的孩子幸福地生活在一起，再也不分离了。
    </body>
    </html>
```

用同样的方法添加另外两个内容页面，分别命名为 content-2.html 和 content-3.html，具体的代码参见代码包中相对应的文件夹。

完成这几个文件之后，打开框架集页面，效果如图 8-23 所示。



图 8-23　框架集页面

单击页面中的链接文字“神笔马良”，会在框架的下方页面中显示 content-2.html 文件的内容，如图 8-24 所示。

单击页面中的链接文字“菊花仙子”，会在框架的下方页面中显示 content-3.html 文件的内容，如图 8-25 所示。这就说明，在框架集中实现了各个窗口页面之间的链接。



图 8-24　链接到 content-2 页面



图 8-25　链接到 content-3 页面

8.5.2　内联框架与链接

在内联框架中同样也可以制作页面之间的链接，其方法与普通框架类似，下面依然通过实例来说明。内联框架所在的主页面命名为 float1.html，代码如下。

【例 8.26】（实例文件：ch08\float1.html）

```
<html>
<head>
<title>内联框架窗口的链接</title>
</head>
<body>
<center>
  <img src="图 05.jpg"><br><br>
```

```
        <a href="content-1.html" target="content">白蛇传</a>
                         
        <a href="content-2.html" target="content">神笔马良</a>
                         
        <a href="content-3.html" target="content">菊花仙子</a>
        <br><br>
    </center>
    <iframe src="content-1.html" name="content" width="500" height="170" align="center">
    </iframe>
    </body>
    </html>
```

在 IE 9.0 中浏览效果如图 8-26 所示。



图 8-26　内联框架窗口的主页面

单击其中的链接文字“菊花仙子”，可以在下面的内联框架页面中打开 content-3.html 文件的内容，如图 8-27 所示。由此可见，实例中成功地实现了内联框架窗口的链接。



图 8-27　链接 content-3.html 文件的效果

8.6 专家解惑

1.如何避免别人把自己的网页放在其框架中

一些网页设计者往往喜欢偷窃别人的劳动成果，比如把别人精心制作的网页，以子页的形式放到自己的框架中。为了尽量避免自己的网页内容被“盗用”，可在网页源代码的<head></head>之间加入如下代码内容。

```
<script language="javascript"><!--
if (self!=top){top.location=self.location;}
--></script>
```

将上述代码加入之后，就可以有效保护自己的网页不被别人放到框架中使用。

2. 如何优化网页布局框架？

框架结构或称帧结构（Frame）网页表现为一个网页内的某一块保持固定，其他部分信息可以通过滚动条上下或左右移动显示。框架型网页的另一个表现是，深层页面的域名通常不会在 URL 中体现出来，即使进入深层子页面，浏览器显示出来的 URL 仍是主页的 URL。

框架结构的好处是：页面整体一致，更新方便。但大多搜索引擎无法识别框架，也没兴趣抓取其中内容。某些浏览器不支持框架页面。

如果要使用框架，必须在代码中使用 Noframes 标签进行优化，把 Noframe 标签看作一个普通文本内容的主页。在<Noframe></Noframe>区域中包含指向 frame 页的链接以及带有关键词的描述文本。同时在框架以外的区域也出现关键词文本。这样搜索引擎才能正确索引到框架内信息。

还有一个办法是采用 iframe 即内联框架（Inner Frame）技术来避免 Frame 带来的不便。所谓 iframe 也是框架的一种形式，它是相当于在主浏览器窗口内嵌一个子窗口，内容自动打开。Iframe 可以嵌在网页中的任意部分，也可以随意定义其大小，其代码显示为：

```
<iframe src=xx width=x height=x scrolling=xx frameborder=x></frame>
```

对搜索引擎来说，iframe 中的文字是可见的，也可以跟踪到其中链接指向的页面，不过与用户所见不同的是，搜索引擎将 iframe 内容看成单独的一个网面内容，与被内嵌的页面无关。

第 9 章

◂ 网页多媒体的设计 ▸

多媒体是一个网站的必备元素，使用它可以丰富网站内容，吸引用户的注意，突出重点。通常多媒体元素包括声音和动画两部分。

9.1　添加动态文字

网页的多媒体元素一般包括动态文字、动态图像、声音以及动画等，其中最简单的就是添加一些滚动文字。

9.1.1　设置滚动文字

使用 marquee 标记可以将文字设置为动态滚动的效果。其语法结构如下。

```
<marquee>滚动文字</marquee>
```



只要在标记之间添加要进行滚动的文字即可，而且可以在标记之间设置这些文字的字体、颜色等。

【例 9.1】（实例文件：ch09\9.1.html）

```
<html>
<head>
<title>设置滚动文字</title>
</head>
<body>
<marquee>
<font face="隶书" color="#CC0000 " size=4>你好，欢迎光临五月蔷薇女裤专卖店!这里有最适合你的打底裤，这里有最让你满意的服务</font>
</marquee>
</body>
</html>
```

在 IE 9.0 中浏览效果如图 9-1 所示。可以看到设置为红色隶书的文字从浏览器的右方缓缓

向左滚动。

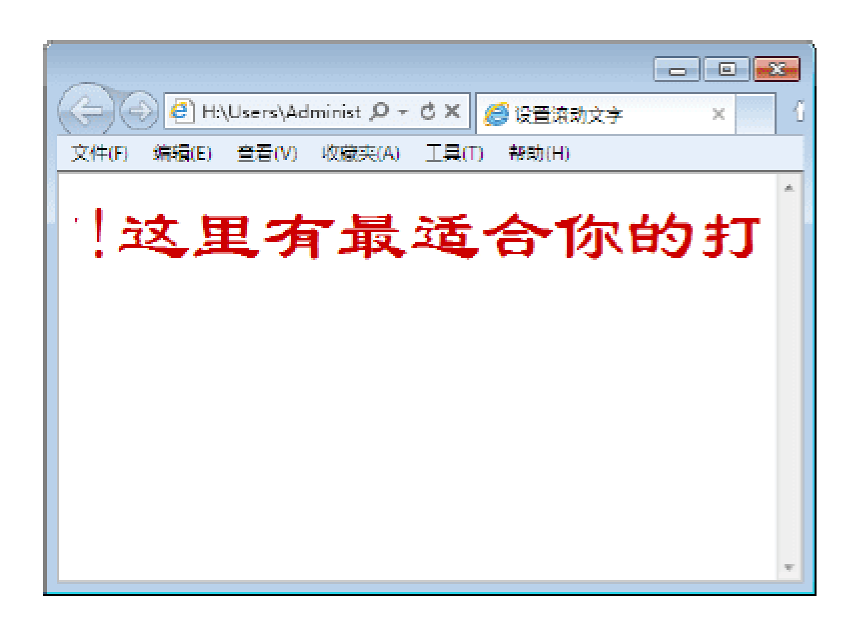


图 9-1　设置滚动文字

9.1.2　设置文字的滚动方向

在网页中除了可以添加滚动的文字外，还可以设置文字的滚动方向，默认情况下文字只能从右向左滚动，而在实际应用中常常需要不同滚动方向的文字，这时就可以通过 direction 参数来设置。语法结构如下。

```
<marquee direction="滚动方向">滚动文字</marquee>
```



该语法中的滚动方向可以包含 4 个取值，分别为 up、down、left 和 right，它们分别表示文字向上、向下、向左和向右滚动，其中向左滚动 left 的效果与默认效果相同，而向上滚动的文字则常常出现在网站的公告栏中。

【例 9.2】（实例文件：ch09\9.2.html）

```
<html>
<head>
<title>设置滚动文字</title>
</head>
<body>
<marquee direction="up">
<font color="#3300FF" face="楷体_GB2312">
清明时节雨纷纷<br>
路上行人欲断魂<br>
借问酒家何处有<br>
牧童遥指杏花村<br>
</font>
</marquee>
<marquee direction="down">
<font color="#FF0000" face="楷体_GB2312">
清明时节雨纷纷<br>
路上行人欲断魂<br>
```

```
借问酒家何处有<br>
牧童遥指杏花村<br>
</font>
</marquee>
</body>
</html>
```

在 IE 9.0 中浏览效果如图 9-2 和图 9-3 所示。可以看到文字不同的滚动效果，图中两段文字的感觉就像是从中间被拉扯开一样。

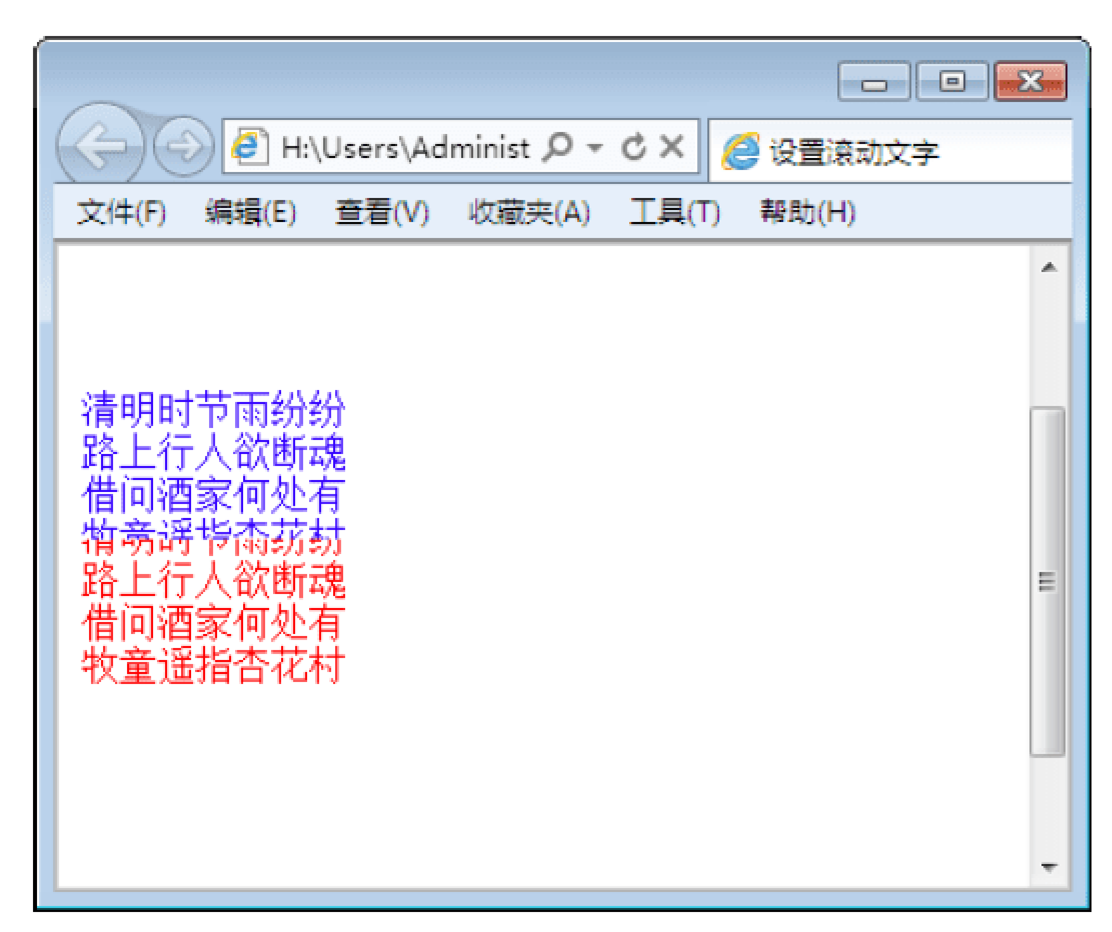


图 9-2 设置滚动方向

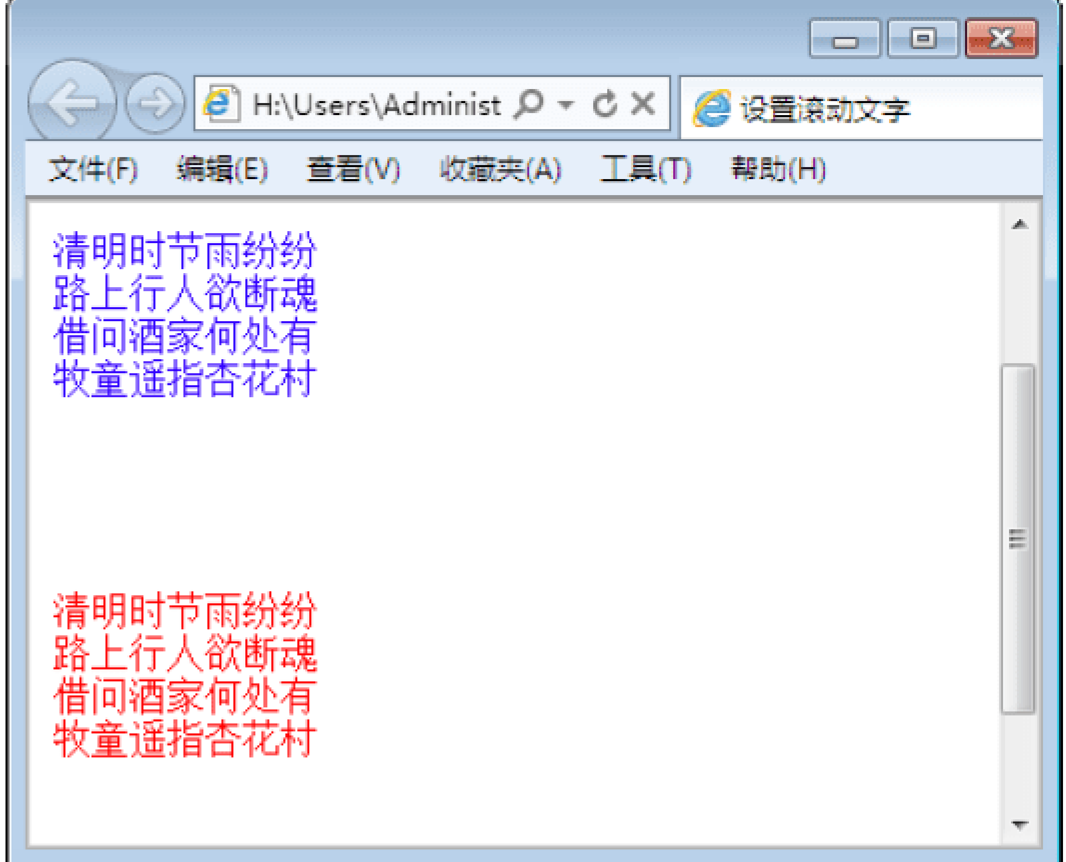


图 9-3 运行一段时间的效果



如果将文字方向相反设置，也就是上面一段文字的 direction 设置为 down，下面一段设置为 up，运行结果就似乎是两组文字融合到一起。

下面设置文字的左右移动效果。

【例 9.3】（实例文件：ch09\9.3.html）

```
<html>
<head>
<title>设置滚动文字</title>
</head>
<body>
<marquee direction=left>啦啦啦，我从右向左移！</marquee> <P>
<marquee direction=right>啦啦啦，我从左向右移！</marquee></font>
</body>
</html>
```

在 IE 8.0 中浏览效果如图 9-4 所示。

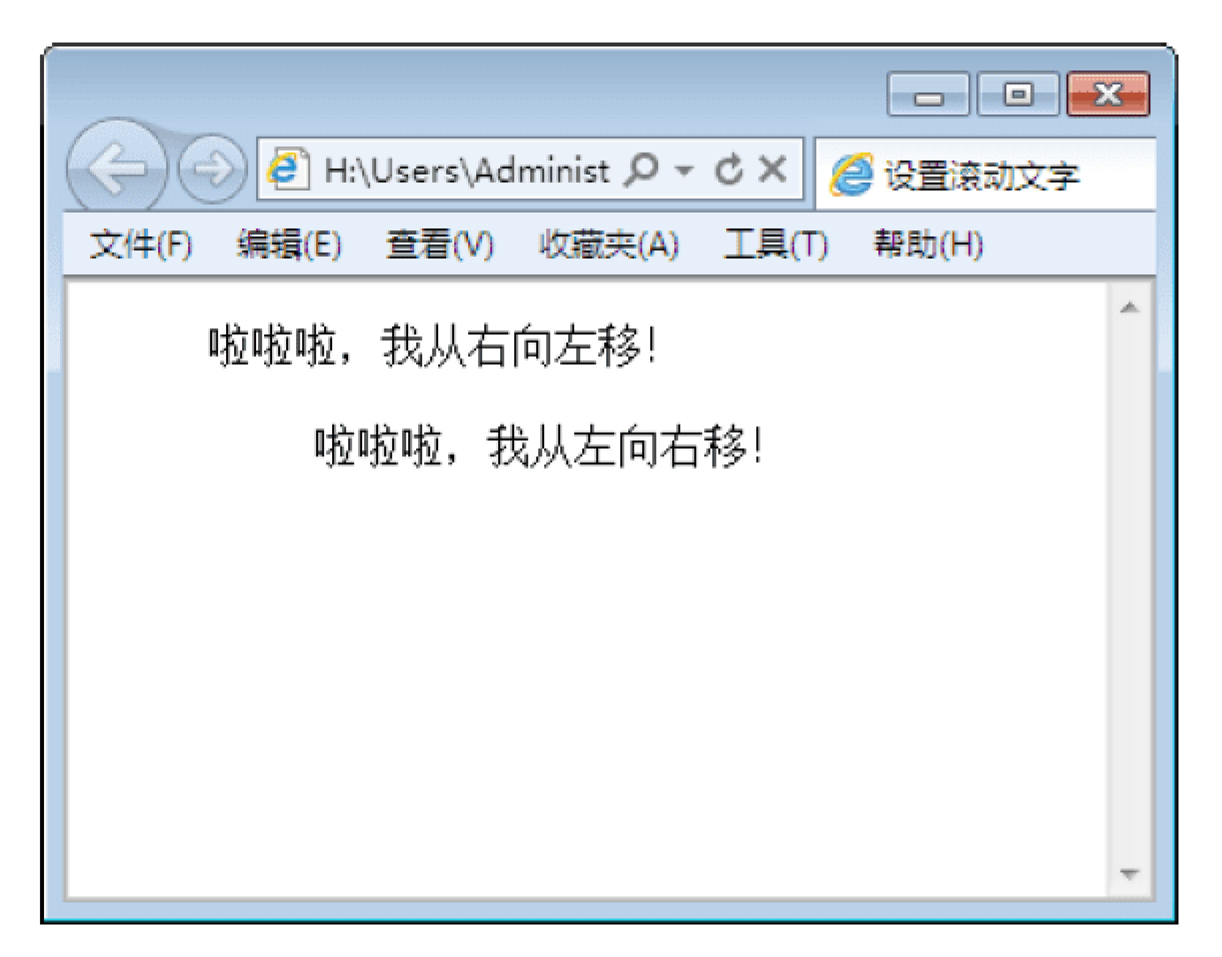


图 9-4　设置文字左右移动的效果

9.1.3　设置文字的滚动方式

除了将文字设置为单方向的滚动外，还可以为文字设置滚动方式，如往复滚动等。这一功能可以通过添加 behavior 属性来实现。语法结构如下。

```
<marquee behavior="滚动方式">滚动文字</marquee>
```

滚动方式 behavior 的取值如表 9-1 所示。

表 9-1　behavior 的取值及效果

取值	滚动的效果
scroll	循环滚动，默认效果
slide	只滚动一次就停止
alternate	来回交替进行滚动

【例 9.4】（实例文件：ch09\9.4.html）

```
<html>
<head>
<title>设置滚动文字</title>
</head>
<body>
marquee behavior=scroll>啦啦啦，我一圈一圈绕着走！</marquee> <P>
<marquee behavior=slide>啦啦啦，我只走一次就歇了！</marquee> <P>
<marquee behavior=alternate>啦啦啦，我来回走耶！</marquee>
</body>
</html>
```

在 IE 9.0 中浏览效果如图 9-5 所示。其中第一行文字不停地循环滚动显示；第二行文字则

在第一次到达浏览器边缘时就停止了滚动；最后一行文字则在滚动到浏览器左边缘后开始反方向运动。

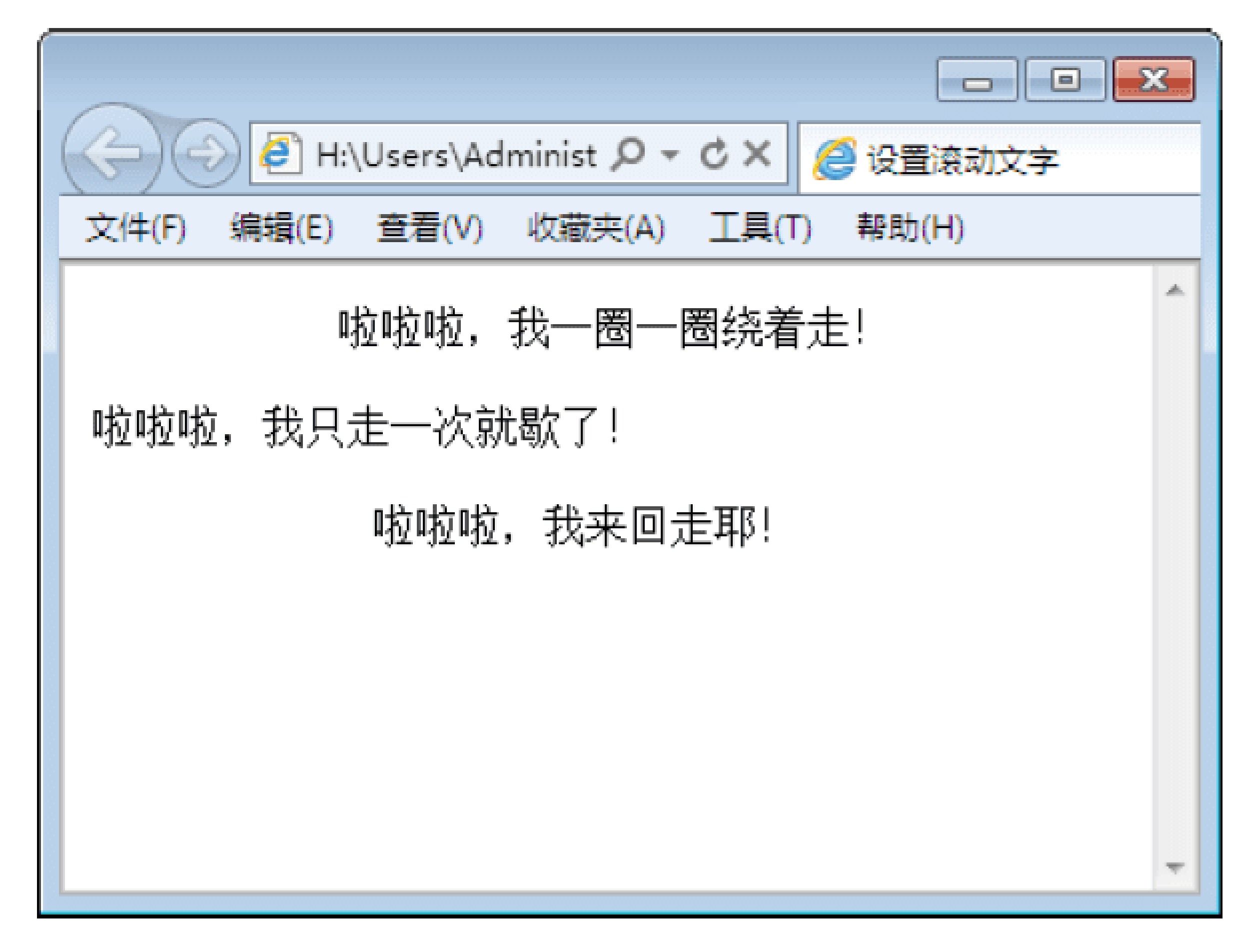


图 9-5　设置滚动方式

9.1.4　设置文字的滚动循环次数

设置滚动文字后，在默认情况下会不断地循环下去，如果希望文字滚动几次后停止，可以使用 loop 参数来进行设置。语法结构如下：

```
<marquee loop="循环次数">滚动文字</marquee>
```

【例 9.5】（实例文件：ch09\9.5.html）

```
<html>
<head>
<title>设置滚动文字</title>
</head>
<body>
<marquee loop=3 behavior=scroll>啦啦啦，我只走 3 趟哟！</marquee> <P>
<marquee loop=3 behavior=slide>啦啦啦，我只走 3 趟哟！</marquee> <P>
<marquee loop=3 behavior=alternate>啦啦啦，我只走 3 趟哟！</marquee>
</body>
</html>
```

在 IE 9.0 中浏览效果如图 9-6 所示。可以看到第一行文字滚动 3 个循环之后，滚动文字将不再出现，第二行与第三行文字，在滚动 3 个循环之后，文字将停留在窗口中。

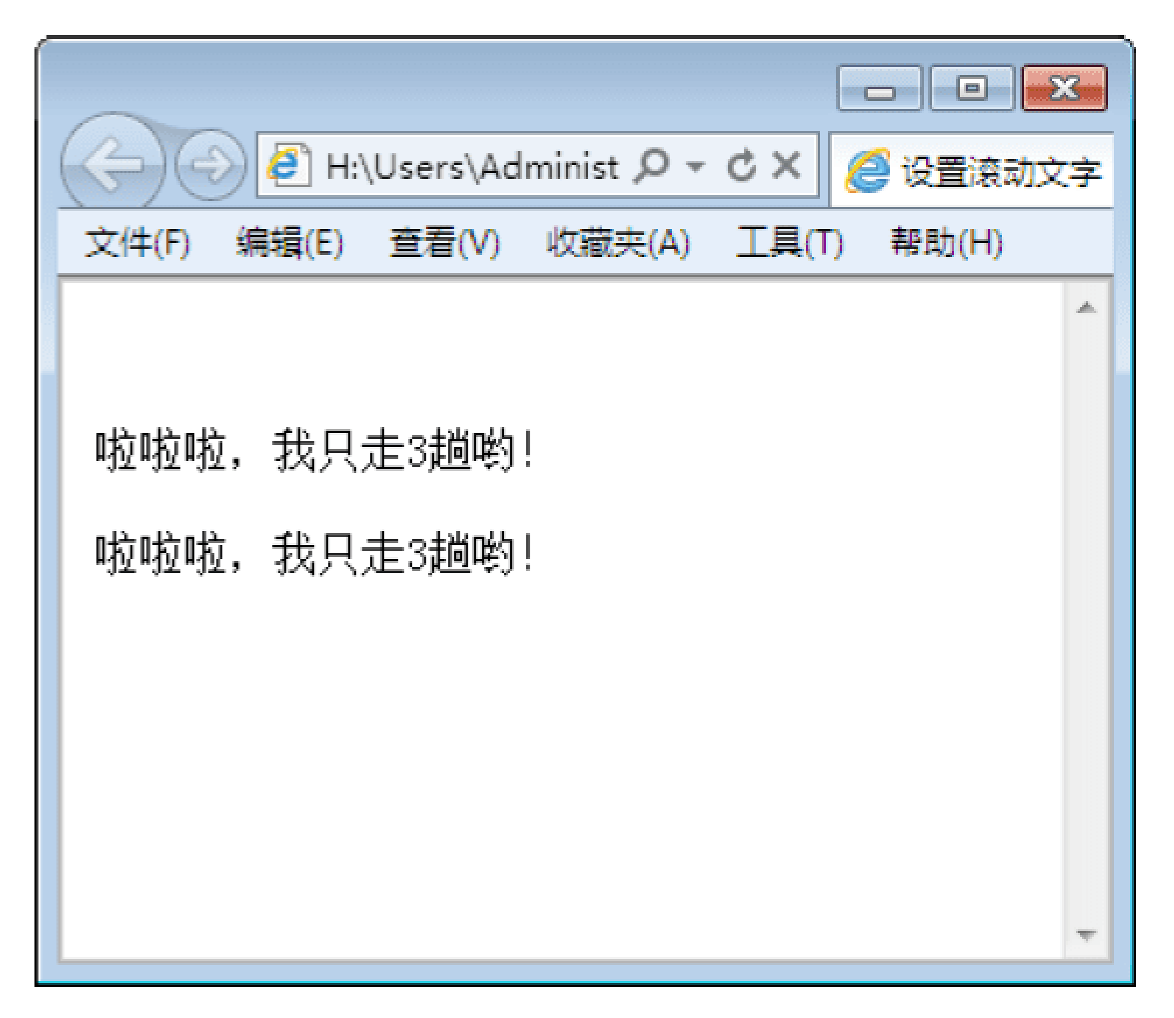


图 9-6 设置滚动的循环次数

9.1.5 设置文字的滚动速度

在设置滚动文字时，有时候希望文字的滚动速度快些，有时则希望慢些，这就需要使用 scrollamount 参数来实现。语法结构如下。

```
<marquee scrollamount=滚动速度></marquee>
```



在该语法中，滚动文字的速度实际上是设置滚动文字每次移动的长度，以像素为单位。

【例 9.6】（实例文件：ch09\9.6.html）

```
<html>
<head>
<title>设置滚动文字</title>
</head>
<body>
<marquee scrollamount=3>哎呀呀，我走的好慢啊！</marquee><br><br>
<marquee scrollamount=10>哎呀呀，我不慌不忙地走啊！</marquee><br><br>
<marquee scrollamount=50>哎呀呀，我走的好快哦！</marquee>
</body>
</html>
```

在 IE 9.0 中浏览效果如图 9-7 所示。可以看到 3 行文字同时开始滚动，但速度是不一样的，设置的 scrollamount 值越大，速度也就越快。

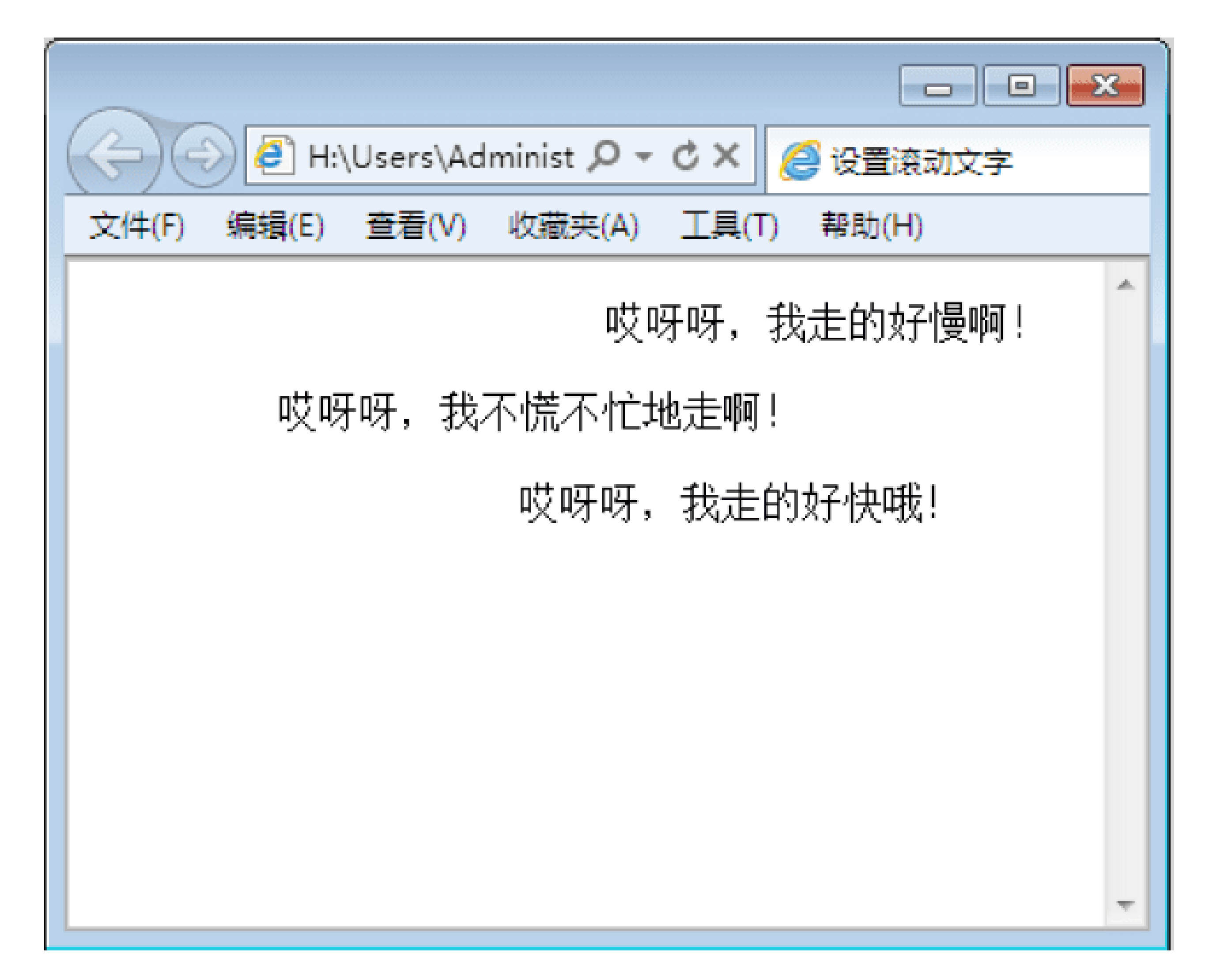


图 9-7　设置文字滚动速度

9.1.6　设置文字滚动的时间间隔

scrolldelay 参数可以设置滚动文字滚动的时间间隔。语法结构如下。

```
<marquee scrolldelay=时间间隔></marquee>
```



scrolldelay 的时间间隔单位是毫秒，也就是千分之一秒。这一时间间隔的设置为滚动两步之间的时间间隔，如果设置的时间比较长，会产生走走停停的效果。如果与滚动速度 scrollamount 参数结合使用，效果会更明显。

【例 9.7】（实例文件：ch09\9.7.html）

```
<html>
<head>
<title>设置滚动文字</title>
</head>
<body>
<marquee scrollamount=10 scrolldelay =5>看我不停脚步地走！ </marquee><br><br>
<marquee scrollamount=10 scrolldelay =100>看我走走歇歇！ </marquee><br><br>
<marquee scrollamount=10 scrolldelay =200>我要走一步停一停</marquee>
</body>
</html>
```

在 IE 9.0 中浏览效果如图 9-8 所示。其中第一行文字设置的延迟小，因此走起来比较平滑；最后一行设置的延迟比较大，看上去就像是走一步歇一会儿的感觉。

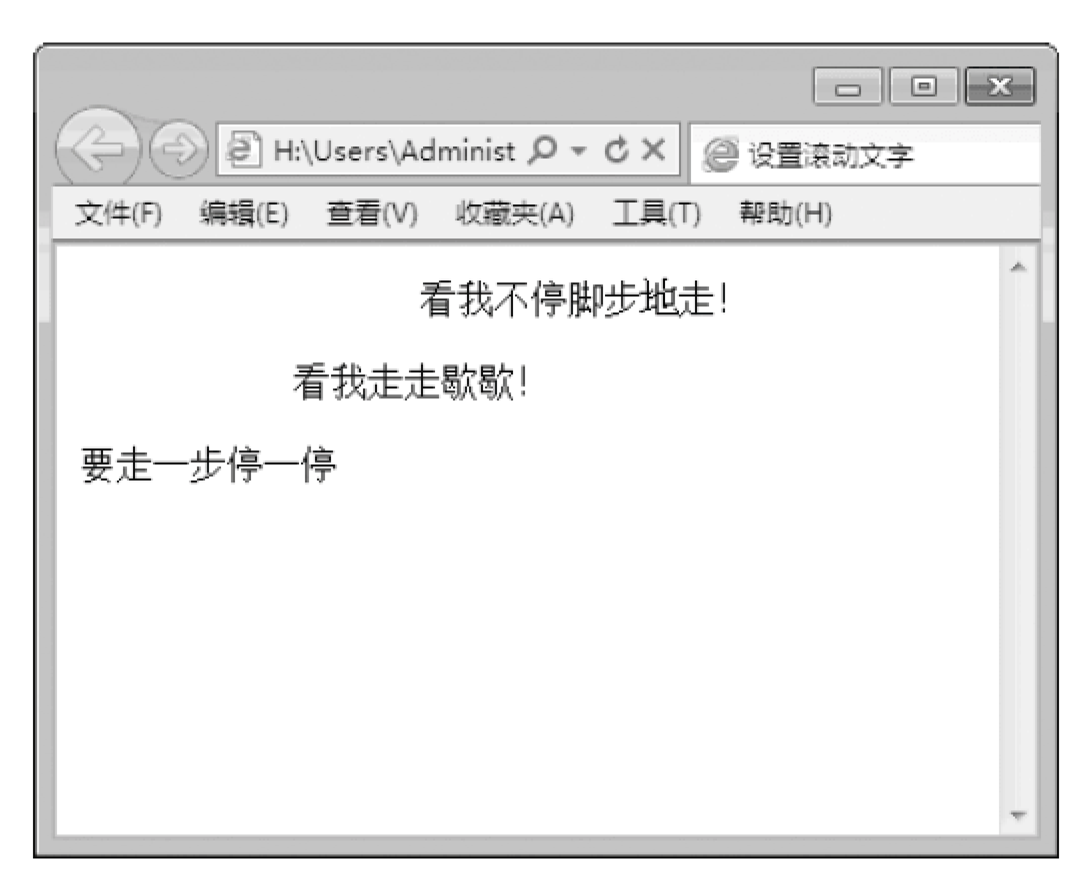


图 9-8　设置文字滚动的时间间隔

9.1.7　设置滚动文字的背景色

在网页中，为了突出某部分内容，常常使用不同背景色来显示。滚动文字也可以单独设置背景色。语法结构如下。

```
<marquee bgcolor="颜色代码">滚动文字</marquee>
```



文字背景颜色设置为 16 位颜色码。

【例 9.8】（实例文件：ch09\9.8.html）

```
<html>
<head>
<title>设置滚动文字</title>
</head>
<body>
<marquee behavior ="alternate" bgcolor="#FFFF66">
沁园春·雪
</marquee>
<br><br>
<marquee direction="up" bgcolor="#99CCFF">
北国风光，千里冰封，万里雪飘。<br>
望长城内外，惟余莽莽；大河上下，顿失滔滔。<br>
山舞银蛇，原驰蜡象，欲与天公试比高。<br>
须晴日，看红装素裹，分外妖娆。<br>
江山如此多娇，引无数英雄竞折腰。<br>
惜秦皇汉武，略输文采；唐宗宋祖，稍逊风骚。<br>
一代天骄，成吉思汗，只识弯弓射大雕。俱往矣，数风流人物，还看今朝。<br>
</marquee>
```

```
</body>
</html>
```

在 IE 9.0 中浏览效果如图 9-9 所示。可以看到在滚动文字后面设置了淡蓝色的背景。

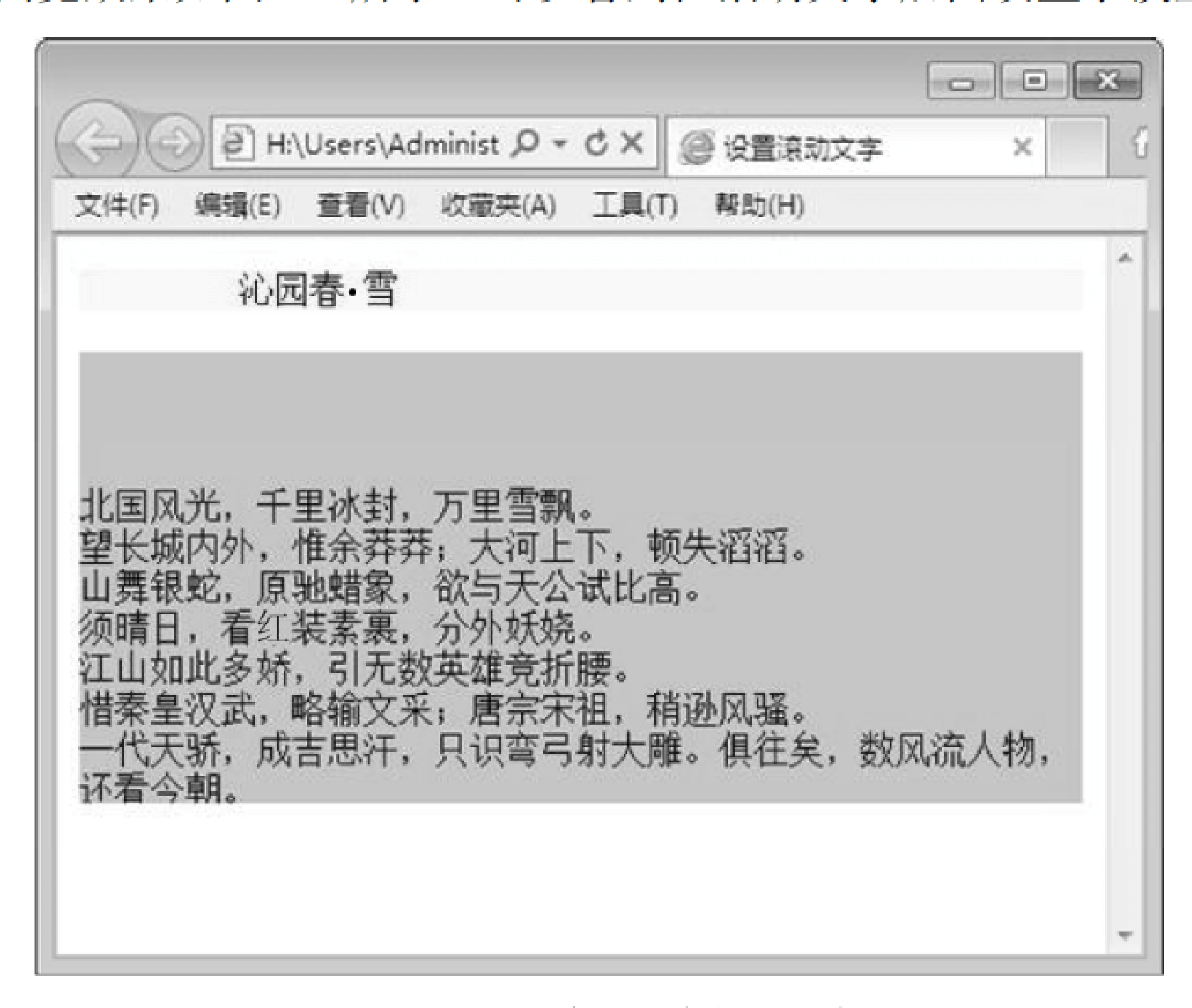


图 9-9　设置滚动文字的背景色

9.1.8　设置滚动文字的背景面积

如果不设置滚动文字的背景面积，那么默认情况下，水平滚动的文字背景与文字同高、与浏览器窗口同宽，使用 width 和 height 参数可以调整其水平和垂直的范围。语法结构如下。

```
<marquee width=背景宽度  height=背景高度>滚动文字</maruquee>
```



此处设置宽度和高度的单位均为像素。

【例 9.9】（实例文件：ch09\9.9.html）

```
<html>
<head>
<title>设置滚动文字</title>
</head>
<body>
<marquee behavior =" alternate" bgcolor="#99CCFF">
欢迎光临我的小店
</marquee><br><br>
<marquee behavior="alternate" bgcolor="#99CCFF" width=500 height=50>
欢迎光临我的小店
```

```
</marquee>
</body>
</html>
```

在 IE 9.0 中浏览效果如图 9-10 所示。可以看到两段滚动文字的背景高度和宽度的变化。

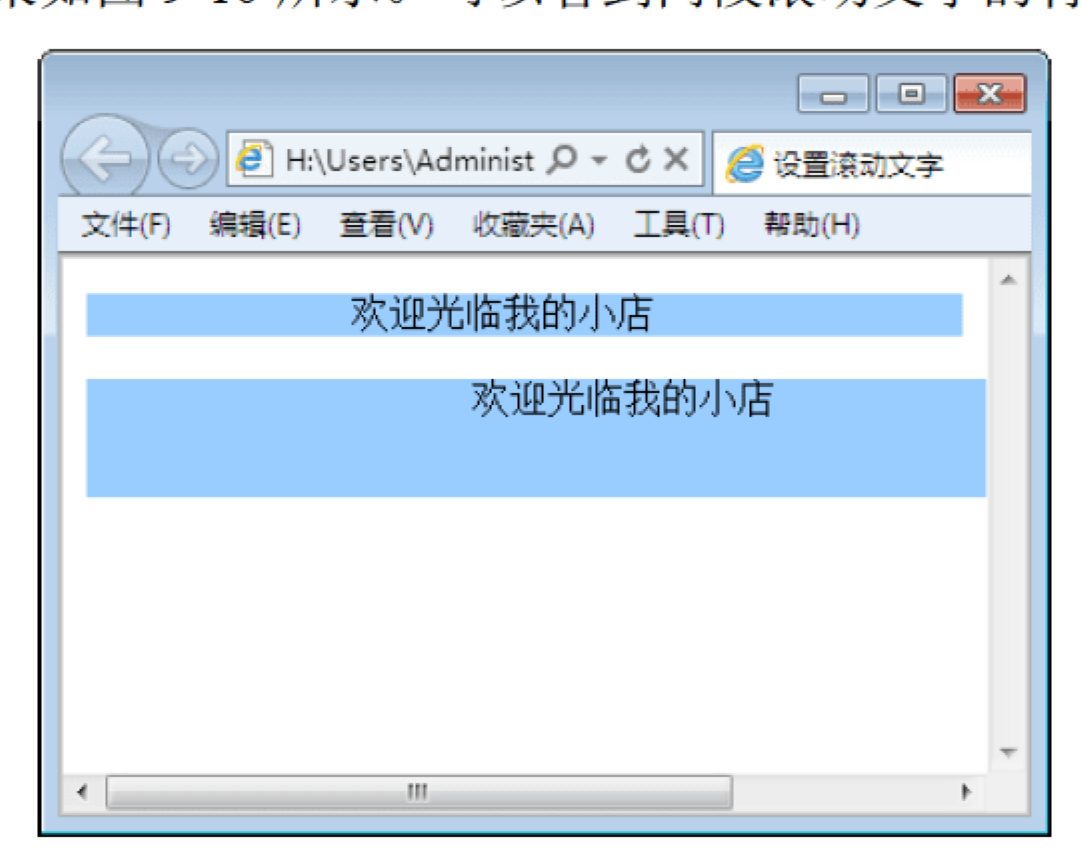


图 9-10　设置滚动背景面积

9.1.9　设置空白空间

默认情况下，滚动文字周围的文字或图像是与滚动背景紧密连接的，使用参数 hspace 和 vspace 可以设置它们之间的空白空间。语法结构如下。

```
<marquee hspace=水平范围  vspace=垂直范围>滚动文字</marquee>
```



该语法中水平和垂直范围的单位均为像素。

【例 9.10】（实例文件：ch09\9.10.html）

```
<html>
<head>
<title>设置滚动文字</title>
</head>
<body>
不设置空白空间的效果：
<marquee behavior ="alternate" bgcolor="#9999FF ">
欢迎光临我的小店
</marquee>
这里有你想象不到的惊喜！
<br>
<hr color="#FF0000">
<br>
设置水平为 70 像素、垂直为 50 像素的空白空间：
```

```
<marquee behavior ="alternate" bgcolor="#9999FF " hspace=70 vspace=50>
欢迎光临我的小店
</marquee>
这里有你想象不到的实惠！
</body>
</html>
```

在 IE 9.0 中浏览效果如图 9-11 所示。可以看到设置空白空间的效果。

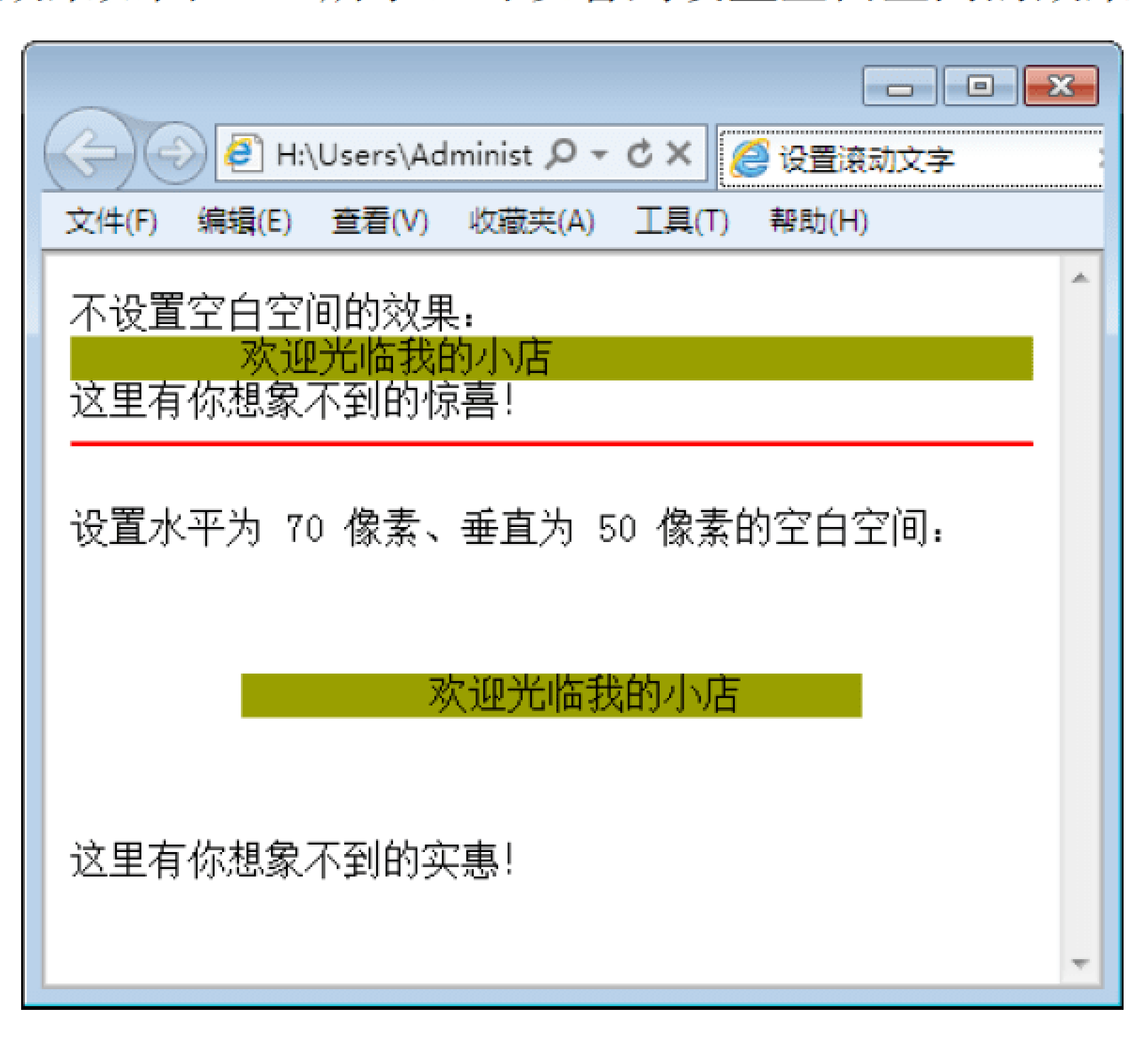


图 9-11 设置空白空间

9.2 添加背景音乐

在 HTML 中，除可以添加动态滚动文字之外，还可以添加背景音乐。

9.2.1 设置背景音乐

在网页中，除可以嵌入普通的声音文件之外，还可以为某个网页设置背景音乐。作为背景音乐的可以是音乐文件，也可以是声音文件，其中最常用的是 midi 文件。语法结构如下。

```
<bgsound src=背景音乐的地址>
```



作为背景音乐的文件还可以是 avi 文件、mp3 文件等声音文件。

【例 9.11】（实例文件：ch09\9.11.html）

```
<html>
<head>
<title>设置背景音乐</title>
```

```
</head>
<body>
<bgsound src="sleep.mp3">
<center>
<font face="黑体" size=5>醉花阴</font><br>
<font size=4>李清照</font>
</center>
<hr width=85% size=3>
<p>    薄雾浓云愁永昼，瑞脑消金兽。<br>
    佳节又重阳，玉枕纱厨，半夜凉初透。<br>
    东离把酒黄昏后，有暗香盈袖。<br>
    莫道不消魂，帘卷西风，人比黄花瘦。</p>
<hr size=5>
</body>
</html>
```

在 IE 9.0 中浏览效果如图 9-12 所示。在打开界面之后就可以听到背景音乐。

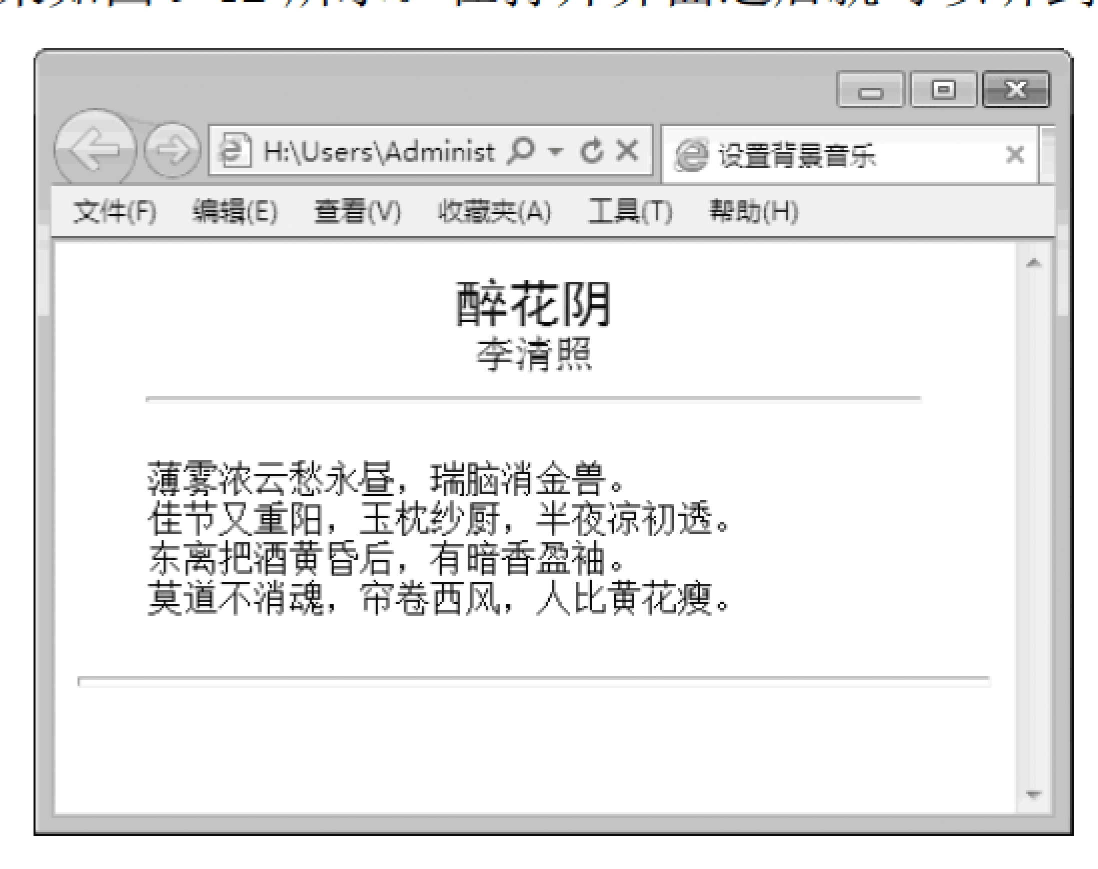


图 9-12 设置背景音乐

9.2.2 设置背景音乐的循环播放次数

通常情况下，背景音乐需要不断地播放，但有时也需要指定播放的次数。使用 loop 参数可以设置背景音乐播放的次数。语法结构如下。

```
<bgsound src=背景音乐的地址 loop=循环次数>
```



如果希望无限次循环播放背景音乐，可将循环次数设置为 true。

【例 9.12】（实例文件：ch09\9.12.html）

```
<html>
<head>
```

```
<title>设置背景音乐</title>
</head>
<body>
<bgsound src="sleep.mid" loop=3>
<center>
<font face="黑体" size=5>醉花阴</font><br>
<font size=4>李清照</font>
</center>
<hr width=85% size=3>
<p>    薄雾浓云愁永昼，瑞脑消金兽。<br>
    佳节又重阳，玉枕纱厨，半夜凉初透。<br>
    东篱把酒黄昏后，有暗香盈袖。<br>
    莫道不消魂，帘卷西风，人比黄花瘦。</p>
<hr size=5>
</body>
</html>
```

在 IE 9.0 中浏览效果，背景音乐在循环播放 3 次以后停止。

9.3 添加多媒体文件

使用 HTML 除了可以为网页添加动态文字、背景音乐外，还可以添加一些多媒体文件，如视频、动画等。这样，可以使单调的网页变得生动。

9.3.1 设置多媒体文件

在网页中常见的多媒体文件包括声音文件和视频文件。设置多媒体文件的语法结构如下。

```
<embed src="多媒体文件地址" width=播放界面的宽度 height=播放界面的高度></embed>
```



在该语法中，width 和 height 一定要设置，单位是像素，否则无法正确显示播放多媒体文件的软件。

【例 9.13】（实例文件：ch09\9.13.html）

```
<html>
<head>
<title>嵌入多媒体文件</title>
</head>
<body>
下面是嵌入的多媒体文件：<br>
<embed src="Wildlife.wmv" width=500 height=200></embed>
</body>
```

```
</html>
```

在 IE 9.0 中浏览效果如图 9-13 所示。单击页面中的播放按钮可以播放插入的文件。



图 9-13 添加多媒体文件

9.3.2 设置多媒体文件的自动运行

登录网页时常常会看到一些视频文件直接开始运行，不需要手动播放，特别是一些广告内容，这是通过 autostart 参数来实现的。语法结构如下：

```
<embed src="多媒体文件地址" autostart=是否自动运行></embed>
```



autostart 的取值有两个。一个是 true，表示自动播放；另一个是 false，表示不自动播放。

【例 9.14】（实例文件：ch09\9.14.html）

```
<html>
<head>
<title>设置自动运行</title>
</head>
<body>
下面的视频文件中上面的视频文件将会自动播放，而下面的视频文件则需要手动播放：<br>
<embed src=" Wildlife.wmv" width=500 height=200 autostart=True></embed>
<embed src=" Wildlife.wmv" width=500 height=200 autostart=False></embed>
</body>
</html>
```

在 IE 9.0 中浏览效果如图 9-14 所示。可以看到两个视频文件的不同效果。



图 9-14 设置自动运行

9.3.3 设置多媒体文件的循环播放

这里的循环播放一般在设置了自动播放时采用，与背景音乐的设置基本相同。语法结构如下：

```
<embed src="多媒体文件地址" loop=是否循环播放></embed>
```



在该语法中，loop 的取值不是具体的数字，而是 true 或者 false，如果取值为 true，表示媒体文件将无限次地循环播放；如果取值为 false，则只播放一次。这里的 loop 也可以设置为播放次数，但是用途并不广泛。

【例 9.15】（实例文件：ch09\9.15.html）

```
<html>
<head>
<title>设置循环播放</title>
</head>
<body>
下面的视频文件将循环播放：<br>
<embed src=" Wildlife.wmv " width=300 autostart=True loop=True></embed>
</body>
</html>
```

在 IE 9.0 中浏览效果如图 9-15 所示。

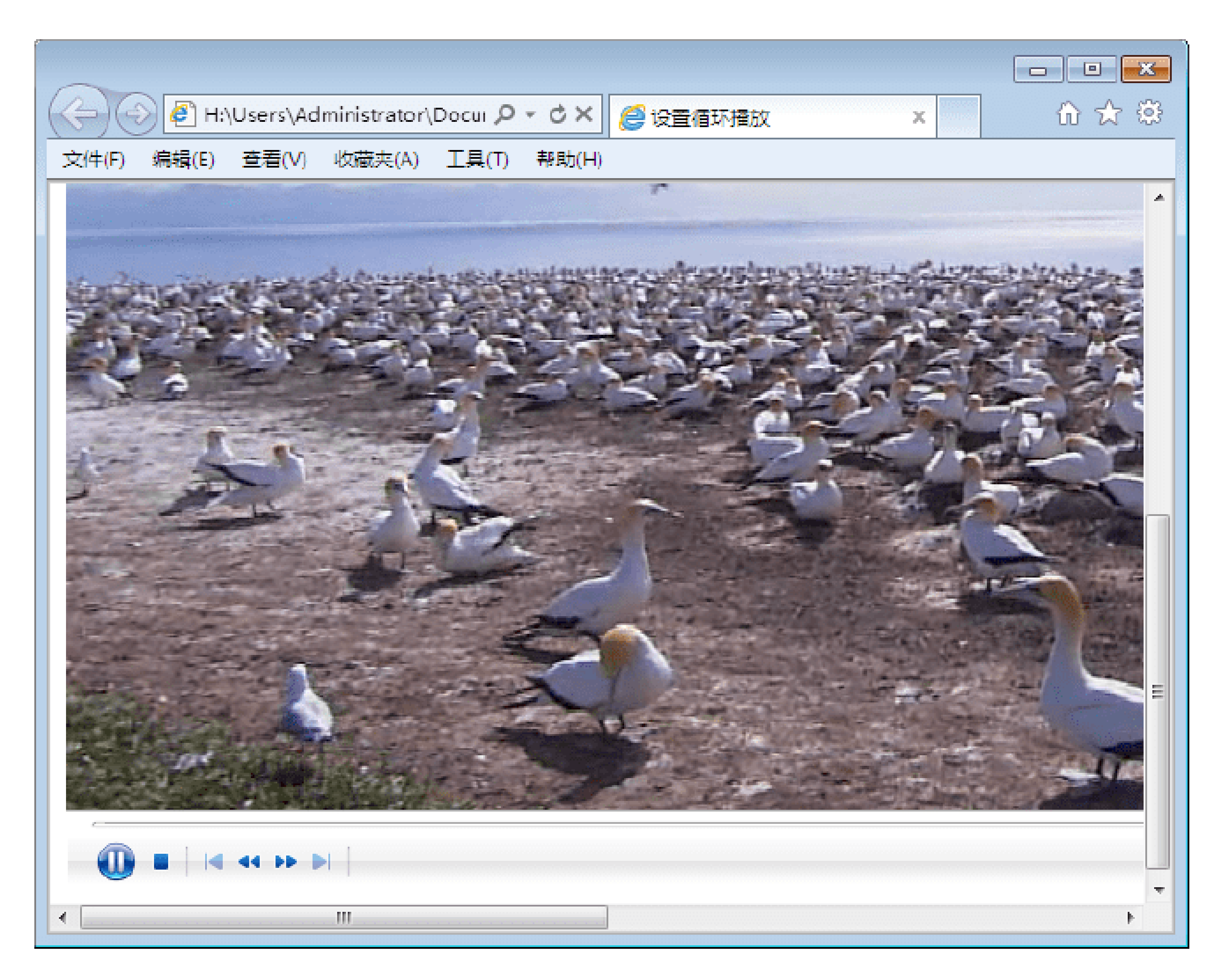


图 9-15 设置媒体的播放次数

9.3.4 隐藏多媒体播放面板

在网页中用户可以将媒体文件的声音保留而隐藏图像，这样就相当于设置了背景声音。通过 hidden 参数可以隐藏播放面板。语法结构如下：

```
<embed src="多媒体文件地址" hidden=是否隐藏></embed>
```



在该语法中，hidden 可以设置为两个值：一个是 true，表示隐藏面板；另一个是 false，表示显示面板，这是添加多媒体文件的默认选项。如果要保留声音，就要设置文件的自动播放。

【例 9.16】（实例文件：ch09\9.16.html）

```
<html>
<head>
<title>设置隐藏面板</title>
</head>
<body>
下面的视频文件播放面板被隐藏：<br>
<embed src=" Wildlife.wmv"width=300 autostart=True hidden=True></embed>
</body>
</html>
```

在 IE 9.0 中浏览效果如图 9-16 所示。可以看到播放控制面板已经不见了，只能听到播放的声音效果。

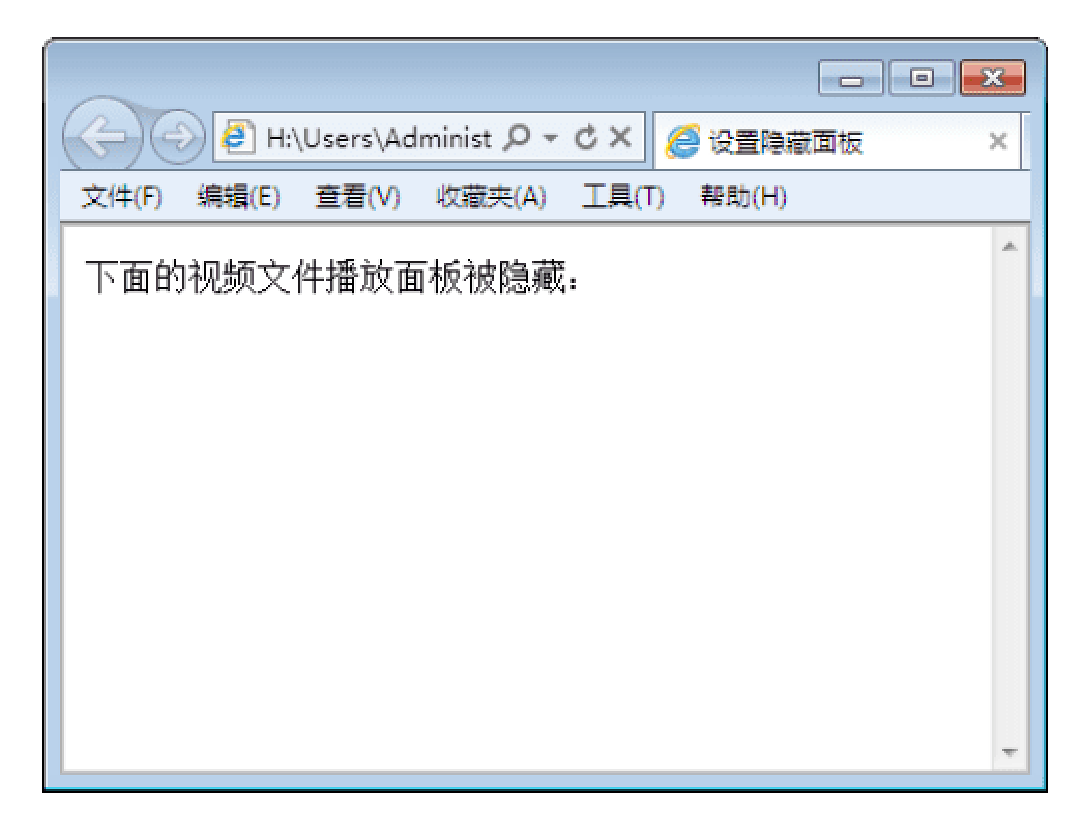


图 9-16 设置隐藏面板

9.3.5 添加其他类型的多媒体文件

除了 wmv 多媒体文件之外，在网页中还可以嵌入 flash、avi 等类型的多媒体文件，方法与 wmv 多媒体文件相同。

【例 9.17】（实例文件：ch09\9.17.html）

```
<html>
<head>
<title>嵌入多媒体文件</title>
</head>
<body>
下面嵌入不同类型的媒体文件：<br>
<embed src="1.swf" width=300></embed>
<embed src="1.avi" width=300 height=200></embed>
</body>
</html>
```

在 IE 9.0 中浏览效果如图 9-17 所示。可以看到在页面中添加了两种不同类型的媒体文件。



图 9-17 添加其他类型的多媒体文件

9.4 专家解惑

1. 如何控制背景音乐的音量？

默认情况下，背景音乐的音量和系统的音量大小是一样的，如果读者可以通过使用 VOLUME 来控制音量的大小，VOLUME 的取值范围为 0~100。例如下面的代码就是控制背景音乐音量大小的例子。

```
<bgsound src="sleep.mp3" VOLUME=50>
```

2. 如何使用<img>标记在网页中插入视频文件？

使用<img>标记在网页中插入视频文件，如下面的代码所示。

```
<img border="0" dynsrc="12.avi" start="fileopen" width="666" height="666">
```

其中 dynsrc 属性用于指定动画文件的文件名，包括文件所在的路径，start 属性用于指定动画开始播放的时间，fileopen 表示网页打开时即播放动画。

但是使用这种标记时，如果用户的电脑上没有相应的视频解码，会出现只有声音、没有图像的现象。

第 10 章

◀ 初识CSS ▶

一个拥有美观、大方、简约的页面以及高访问量的网站，是网页设计者的追求。然而，仅通过 HTML 实现是非常困难的，HTML 语言仅仅定义了网页结构，而对于文本样式没有过多涉及。这就需要一种技术对页面布局、字体、颜色、背景和其他图文效果的实现提供更加精确的控制，这种技术就是 CSS。

10.1　为什么要使用 CSS

随着 Internet 不断发展，对页面效果诉求越来越强烈，只依赖 HTML 这种结构化标记来实现样式已经不能满足网页设计者的需要。其表现有以下几个方面。

- 维护困难，为了修改某个特殊标记格式，需要花费很长的时间，尤其对整个网站而言，后期修改和维护成本较高。
- 标记不足，HTML 本身标记很少，很多标记都是为网页内容服务的，而关于内容样式标记，例如文字间距、段落缩进很难在 HTML 中找到。
- 网页过于臃肿，由于没有同意对各种风格样式进行控制，HTML 页面往往体积过大，占用掉很多宝贵的宽度。
- 定位困难，在整体布局页面时，HTML 对于各个模块的位置调整显得捉襟见肘，过多的 table 标记将会导致页面复杂和后期维护困难。

在这种情况下，就需要寻找一种可以将结构化标记与丰富的页面表现相结合的技术。CSS 样式技术就产生了。

CSS（Cascading Style Sheet）称为层叠样式表，也可以称为 CSS 样式表或样式表，其文件扩展名为.css。CSS 是用于增强或控制网页样式，并允许将样式信息与网页内容分离的一种标记性语言。

引用样式表的目的是将“网页结构代码”和“网页样式风格代码”分离开，从而使网页设计者可以对网页布局进行更多的控制。利用样式表，可以将整个站点上所有网页都指向某个 CSS 文件，设计者只需要修改 CSS 文件中的某一行，整个网页上对应的样式都会随之发生改变。

10.2 在 HTML 中使用 CSS 的方法

CSS 样式表能很好地控制页面显示，以达到分离网页内容和样式代码的目的。CSS 样式表控制 HTML 页面达到好的样式效果，其方式通常包括行内样式、内嵌样式、链接样式和导入样式。

10.2.1 行内样式

行内样式是所有样式中比较简单、直观的方法，就是直接把 CSS 代码添加到 HTML 的标记中，即作为 HTML 标记的属性标记存在。通过这种方法，可以很简单地对某个元素单独定义样式。

使用行内样式方法是直接在 HTML 标记中使用 style 属性，该属性的内容就是 CSS 的属性和值，例如：

```
<p style="color:red">段落样式</p>
```

【例 10.1】（实例文件：ch10\10.1.html）

```
<html>
<head>
<title>行内样式</title>
</head>
<body>
<p style="color:red;font-size:20px;text-decoration:underline;text-align:center">此段落使用行内样式修饰
</p>
<p style="color:blue;font-style:italic">小荷才露尖尖角</p>
<p style="color:blue;font-style:italic">早有蜻蜓立上头</p>
</body>
</html>
```

在 IE 9.0 中浏览效果如图 10-1 所示，可以看到 2 个 p 标记中都使用了 style 属性，并且设置了 CSS 样式，各个样式之间互不影响，分别显示自己的样式效果。第 1 个段落设置红色字体，居中显示，带有下滑线。第二个段落蓝色字体，以斜体显示。



行内样式简单，但这种方法不经常使用，因为这样添加无法完全发挥样式表“内容结构和样式控制代码”分离的优势。而且这种方式也不利于样式的重用。如果需要为每一个标记都设置 style 属性，则后期维护成本高，网页容易过胖，故不推荐使用。

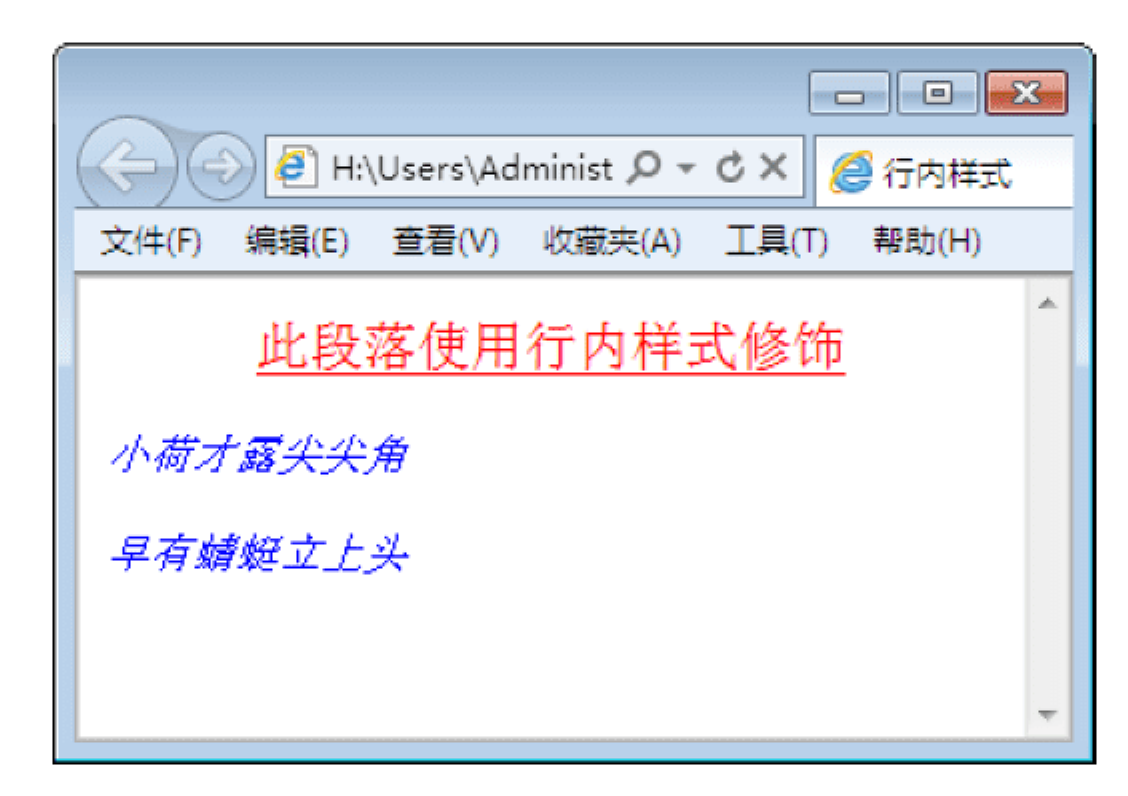


图 10-1　行内样式显示

10.2.2　内嵌样式

内嵌样式就是将 CSS 样式代码添加到<head>与</head>之间，并且用<style>和</style>标记进行声明。这种写法虽然没有完全实现页面内容和样式控制代码完全分离，但可以设置一些比较简单的样式，并统一页面样式。其格式如下所示。

```
<head>
  <style type="text/css" >
    p
    {
      color:red;
      font-size:12px;
    }
  </style>
</head>
```

有些较低版本的浏览器不能识别<style>标记，因而不能正确地将样式应用到页面显示上，而是直接将标记中的内容以文本的形式显示。为了解决此类问题，可以使用 HMTL 注释将标记中的内容隐藏。如果浏览器能够识别<style>标记，则标记内被注释的 CSS 样式定义代码依旧能够发挥作用。

```
<head>
  <style type="text/css" >
  <!--
    p
    {
      color:red;
      font-size:12px;
    }
  -->
  </style>
```

```
</head>
```

【例 10.2】（实例文件：ch10\10.2.html）

```
<html>
<head>
<title>内嵌样式</title>
<style type="text/css">
p{
        color:orange;
        text-align:center;
        font-weight:bolder;
        font-size:25px;
}
</style>
</head>
<body>
<p>此段落使用内嵌样式修饰</p>
<p>采得百花成蜜后</p>
<p>为谁辛苦为谁忙</p>
</body>
</html>
```

在 IE 9.0 中浏览效果如图 10-2 所示，可以看到两个 p 标记都被 CSS 样式修饰，其样式保持一致，段落居中、字体加粗并以橙色显示。

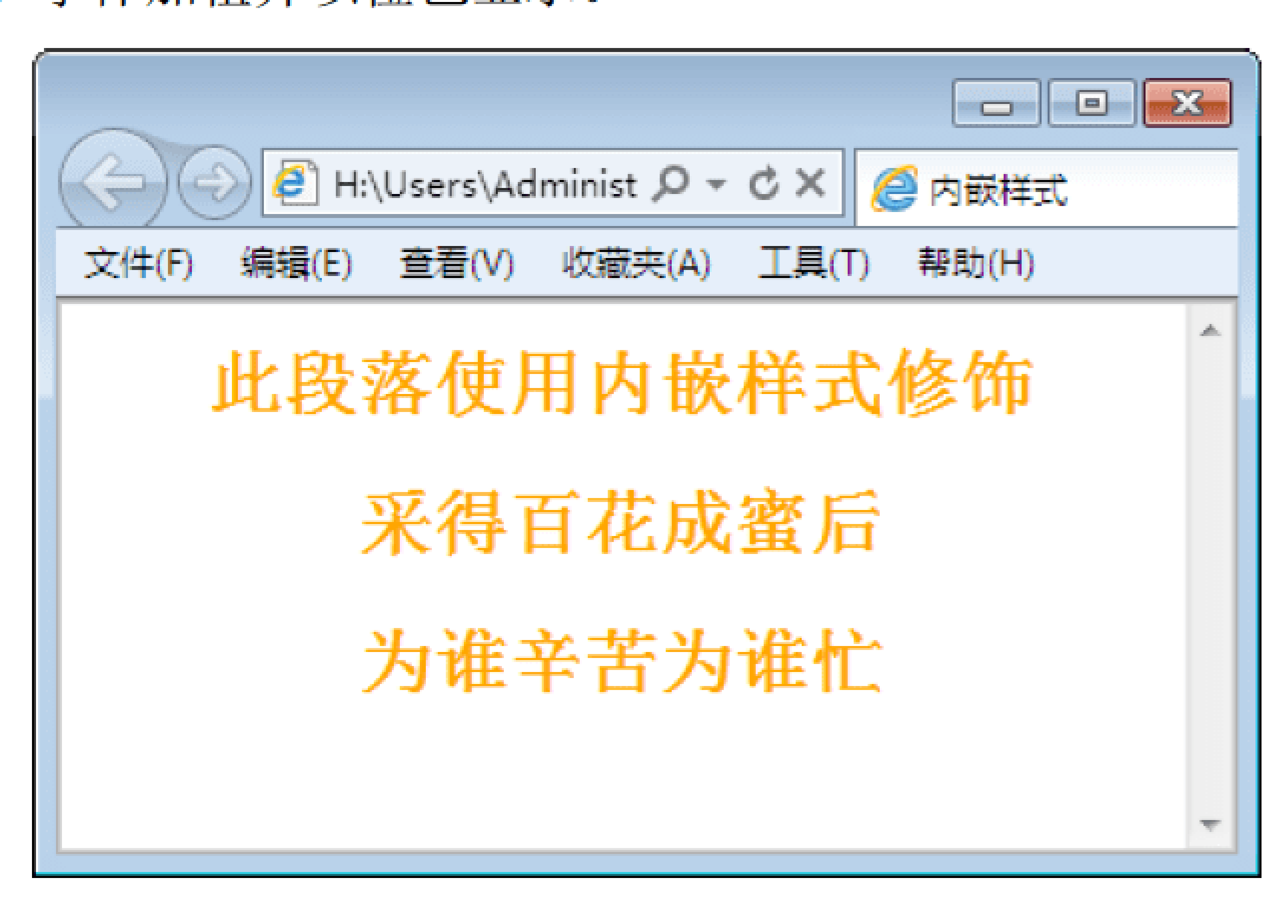


图 10-2　内嵌样式显示

在上面的例子中，所有 CSS 编码都在 style 标记中，方便了后期维护，页面与行内样式相比大大瘦身了。但如果一个网站，拥有很多页面，对于不同页面 p 标记都希望采用同样风格时，内嵌方式就显得有点麻烦。此种方法只适用于对特殊页面设置单独的样式风格。

10.2.3　链接样式

链接样式是 CSS 中使用频率最高，也是最实用的方法。它很好地将“页面内容”和“样式风格代码”分离成两个或多个文件，实现了页面框架 HTML 代码和 CSS 代码的完整分离，使前期制作和后期维护都十分方便。同一个 CSS 文件，根据需要可以链接到网站中所有的 HTML 页面上，使得网站整体风格统一、协调，并且后期维护的工作量也大大减少。

链接样式是指在外部定义 CSS 样式表并形成以.css 为扩展名的文件，然后在页面中通过<link>链接标记链接到页面中，而且该链接语句必须放在页面的<head>标记区，如下所示：

```
<link rel="stylesheet" type="text/css" href="1.css" />
```

- rel 指定链接到样式表，其值为 stylesheet。
- type 表示样式表类型为 CSS 样式表
- href 指定了 CSS 样式表所在位置，此处表示当前路径下名称为 1.css 的文件。

这里使用的是相对路径。如果 HTML 文档与 CSS 样式表没有在同一路径下，则需要指定样式表的绝对路径或引用位置。

【例 10.3】（实例文件：ch10\10.3.html）

```
<html>
<head>
<title>链接样式</title>
<link rel="stylesheet" type="text/css" href="9.3.css" />
</head>
<body>
<h5>此段落使用链接样式修饰</h5>
<p>泉眼无声惜细流, </p>
<p>树阴照水爱晴柔。</p>
</body>
</html>
```

【例 10.4】（实例文件：ch10\10.3.css）

```
h1{text-align:center;}
p{font-weight:29px;text-align:center;font-style:italic;}
```

在 IE 9.0 中浏览效果如图 10-3 所示，标题和段落以不同样式显示，标题居中显示，段落以斜体居中显示。



图 10-3　链接样式显示

链接样式的最大优势就是将 CSS 和 HTML 代码完全分离，并且同一个 CSS 文件能被不同的 HTML 链接所使用。



在设计整个网站时，可以将所有页面链接到同一个 CSS 文件，使用相同的样式风格。如果整个网站需要修改样式，只修改 CSS 文件即可。

10.2.4　导入样式

导入样式和链接样式基本相同，都是创建一个单独的 CSS 文件，然后再引入到 HTML 文件中。只不过语法和运作方式有差别。采用导入样式的样式表，在 HTML 文件初始化时，会被导入到 HTML 文件内，作为文件的一部分，类似于内嵌效果。而链接样式是在 HTML 标记需要样式风格时才以链接方式引入。

导入外部样式表是指在内部样式表的<style>标记中，使用@import 导入一个外部样式表，例如：

```
<head>
  <style type="text/css" >
  <!--
  @import "1.css"
  -->
</style>
</head>
```

导入外部样式表相当于将样式表导入到内部样式表中，其方式更有优势。导入外部样式表必须在样式表的开始部分，其他内部样式表上面。

【例 10.5】（实例文件：ch10\10.5.html）

```
<html>
<head>
<title>导入样式</title>
<style>
@import "9.6.css"
```

```
</style>
</head>
<body>
<h1>CSS 学习</h1>
<p>此段落使用导入样式修饰</p>
</body>
</html>
```

【例 10.6】（实例文件：ch10\10.5.css）

```
h1{text-align:center;color:#0000ff}
p{font-weight:bolder;text-decoration:underline;font-size:20px;}
```

在 Firefox 5.0 中浏览效果如图 10-4 所示，标题和段落以不同样式显示，标题居中显示颜色为蓝色，段落以大小 20px 并加粗显示。



图 10-4　导入样式显示

导入样式与链接样式比较，最大的优点就是可以一次导入多个 CSS 文件，其格式如下所示：

```
<style>
@import "9.5.css"
@import "test.css"
</style>
```

10.2.5　优先级问题

如果同一个页面，采用了多种 CSS 使用方式，例如使用行内样式、链接样式和内嵌样式。如果这几种样式共同作用于同一个标记，就会出现优先级问题，即究竟哪种样式设置有效。例如，如果内嵌设置字体为宋体，链接样式设置为红色，那么两者会同时生效，而如果都设置字体颜色，情况就会变得复杂。

1．行内样式和内嵌样式

例如，有这样一种情况：

```
<style>
```

```
.p{color:red}
</style>
<p style = " color:blue ">段落应用样式</p>
```

在样式定义中，段落标记<p>匹配了两种样式规则，一种使用内嵌样式定义颜色为红色，一种使用 p 行内样式定义颜色为蓝色，而在页面代码中，该标记使用了类选择符。但是，标记内容最终会以哪一种样式显示呢？

【例 10.7】（实例文件：ch10\10.6.html）

```
<html>
<head>
<title>优先级比较</title>
<style>
.p{color:red}
</style>
</head>
<body>
<p style = " color:blue ">优先级测试</p>
</body>
</html>
```

在 IE 9.0 中浏览效果如图 10-5 所示，段落以蓝色字体显示，可以知道行内优先级大于内嵌优先级。

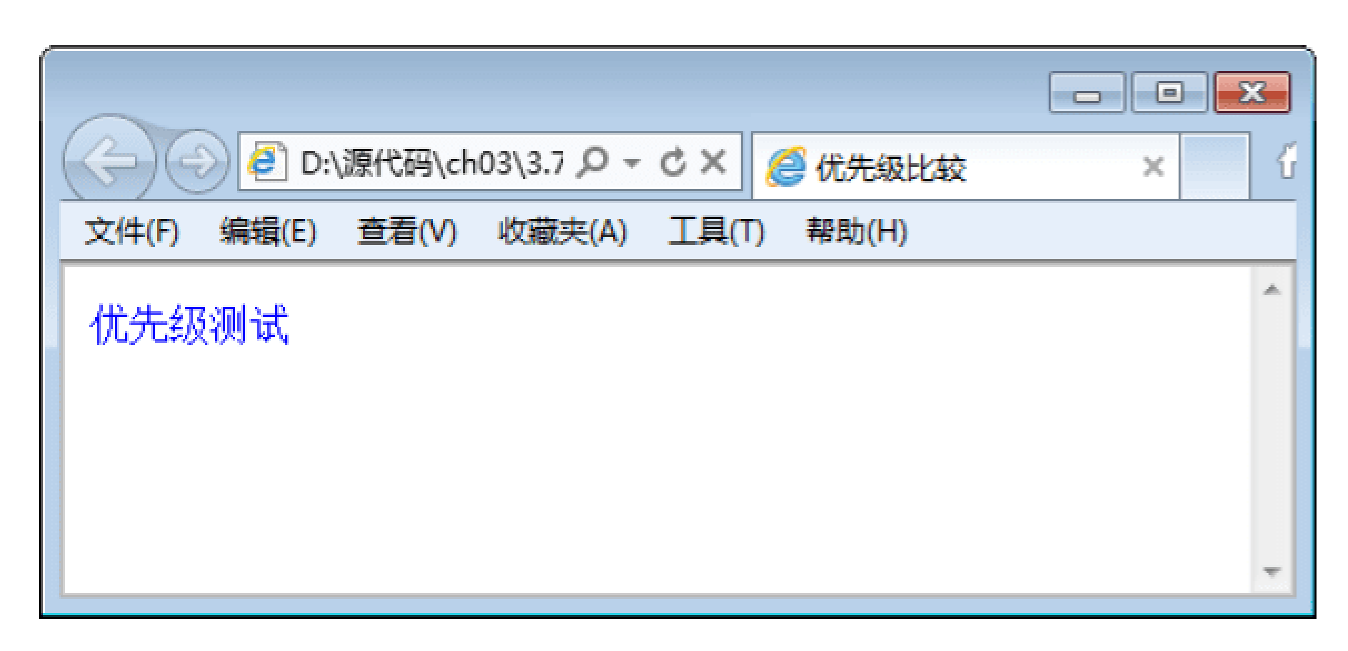


图 10-5 优先级显示

2．内嵌样式和链接样式

以相同例子测试内嵌样式和链接样式优先级，设置颜色样式代码，单独放在一个 CSS 文件中，使用链接样式引入。

【例 10.8】（实例文件：ch10\10.7.html）

```
<html>
<head>
<title>优先级比较</title>
<link href="9.7.css" type="text/css" rel="stylesheet">
```

```
<style>p{color:red}
</style></head>
<body>
<p>优先级测试</p>
</body>
</html>
```

【例 10.9】（实例文件：ch10\10.7.css）

```
p{color:yellow}
```

在 IE 9.0 中浏览效果如图 10-6 所示，段落以红色字体显示。

从上面的代码中可以看出，内嵌样式和链接样式同时对段落 p 修饰，段落显示红色字体。可以知道，内嵌样式优先级大于链接样式。

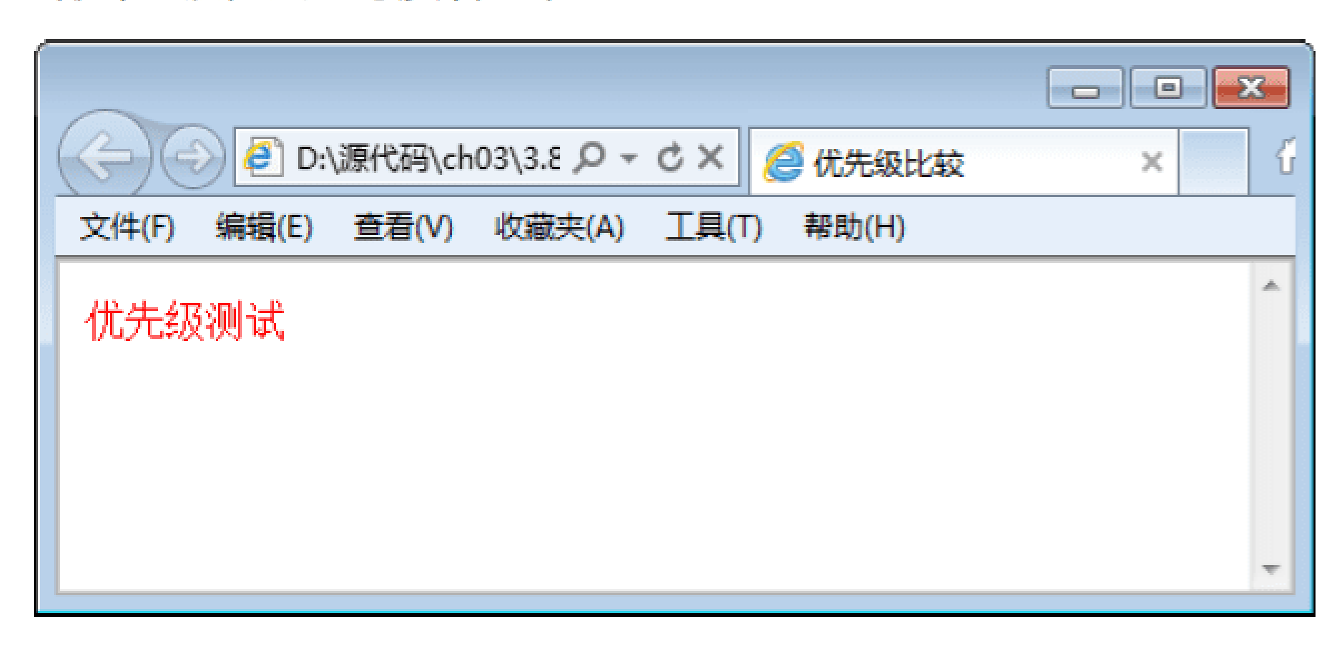


图 10-6　优先级测试

3．链接样式和导入样式

现在进行链接样式和导入样式测试，分别创建两个 CSS 文件，一个作为链接，一个作为导入。

【例 10.10】（实例文件：ch10\10.8.html）

```
<html>
<head>
<title>优先级比较</title>
<style>
@import "9.8_2.css"
</style>
<link href="9.8_1.css" type="text/css" rel="stylesheet">
</head><body>
<p>优先级测试</p>
</body></html>
```

【例 10.11】（实例文件：ch10\10.8_1.css）

```
p{color:green}
```

【例 10.12】（实例文件：ch10\10.8_2.css）

```
p{color:purple}
```

在 IE 9.0 中浏览效果如图 10-7 所示，段落以绿色显示。

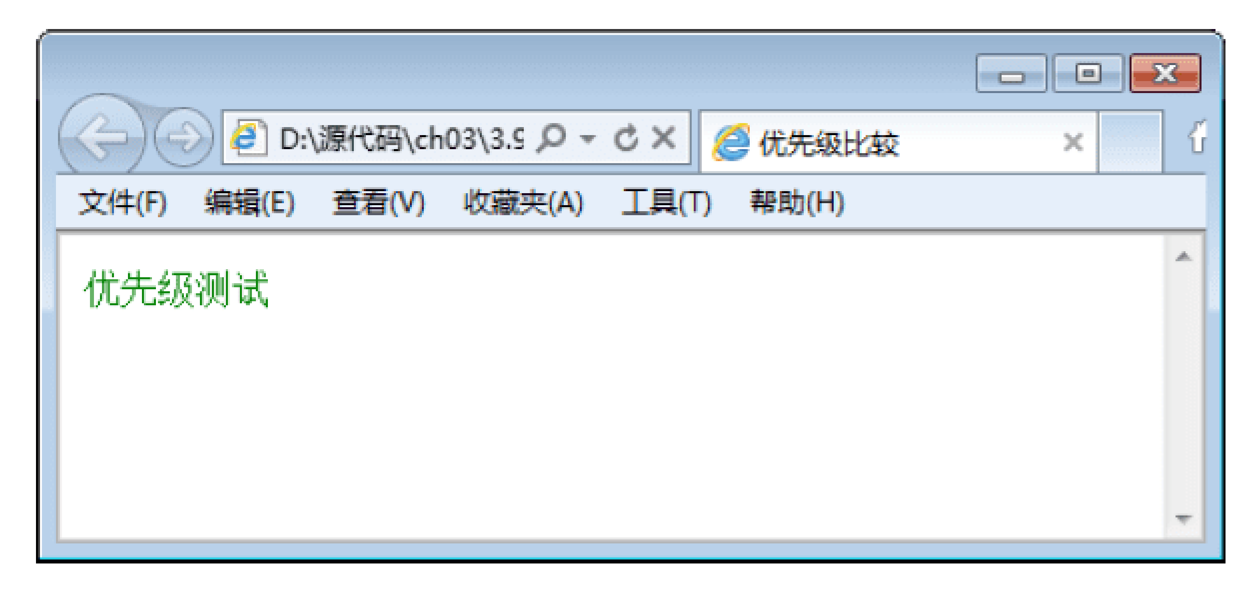


图 10-7　优先级比较

从上面的代码可以看出，此时链接样式优先级大于导入样式优先级。

10.3　专家解惑

1．CSS 定义字体在不同浏览器中大小不一样怎么办？

例如使用 font-size:14px 定义的宋体文字，在 IE 下实际高是 16px，下空白是 3px，在 Firefox 浏览器下实际高是 17px、上空 1px、下空 3px。其解决办法是在文字定义时设定 line-height，并确保所有文字都有默认的 line-height 值。

2．CSS 在网页制作中一般有 4 种用法，那么具体在使用时该采用哪种？

有多个网页要用到的 CSS，采用外连 CSS 文件的方式，这样可大大减少网页的代码，修改起来非常方便；只在单个网页中使用的 CSS，采用文档头部方式；只有在一个网页的一、两个地方才用到的 CSS，采用行内插入方式。

3．CSS 的行内样式、内嵌样式和链接样式可以在一个网页中混用吗？

3 种用法可以混用，且不会造成混乱。这就是它为什么称为“层叠样式表”的原因，浏览器在显示网页时是这样处理的：先检查有没有行内插入式 CSS，有就执行，针对本句的其他 CSS 就不去管它了；其次检查内嵌方式的 CSS，有就执行；在前两者都没有的情况下再检查外连文件方式的 CSS。因此可看出，3 种 CSS 的执行优先级是：行内样式、内嵌样式、链接样式。

第 11 章

CSS基础语法

通过使用 CSS 样式规则能够修饰 HTML 标记，起到美化网页的作用。但其更大的作用是将网页内容和网页样式二者分离，这样方便了后期维护。本章将介绍 CSS 的基础语法和使用。

11.1 CSS 选择器

选择器（selector）也称为选择符。HTML 语言中的所有标记都是通过不同的 CSS 选择器进行控制的。选择器不只是 HMTL 文档中的元素标记，它还可以是类（Class，这不同于面向对象中的类）、ID（元素的唯一特殊名称，便于在脚本中使用）或是元素的某种状态（如 a:link）。根据 CSS 选择符用途可以把选择器分为标签选择器、类选择器、全局选择器、ID 选择器和伪类选择器等。

11.1.1 标签选择器

HTML 文档由多个不同标记组成，而 CSS 选择器就用于声明那些标记采用样式。例如，p 选择器就用于声明页面中所有<p>标记的样式风格。同样也可以通过 h1 选择器来声明页面中所有<h1>标记的 CSS 风格。

标签选择器的基本形式如下所示：

```
tagName{property:value}
```

其中 tagName 表示标记名称，例如 p、h1 等 HTML 标记；proerty 表示 CSS 属性；value 表示 CSS 属性值。

通过一个具体标记来命名，可以对文档里该标记出现的地方应用样式定义。这种做法通常用于设置那些在整个网站都会出现的基本样式。例如，下面的代码就用于为一个网站设置默认字体。

```
body, p, td, th, div, blockquote, dl, ul, ol {
   font-family: Tahoma, Verdana, Arial, Helvetica, sans-serif;
   font-size: 1em;
   color: #000000;
}
```

此选择器是一系列的标记，所有这些标记都将以定义的样式（字体、字号和颜色）显示。理论上，〈body〉标记就是所需要的全部（因为所有其他标记都会出现在〈body〉标记内部，并且将因此继承它的属性），但是许多浏览器都不能恰当地将这些样式属性带入表格和其他元素里。因此，为了完整，这里指定了其他元素。

【例 11.1】（实例文件：ch11\11.1.html）

```
<html>
<head>
<title>标签选择器</title>
<style>
    p{color:blue;font-size:20px;}
</style>
</head>
<body>
    <p>此处使用标签选择器控制段落样式</p>
</body>
</html>
```

在 IE 9.0 中浏览效果如图 11-1 所示，可以看到段落以蓝色字体显示，大小为 20px。

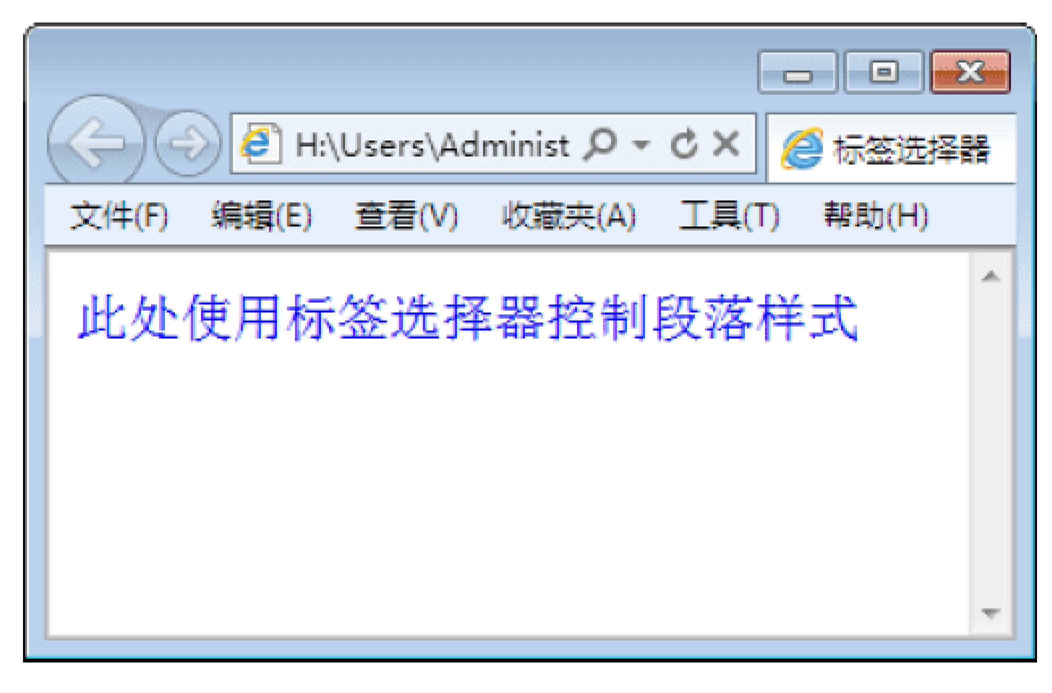


图 11-1　标签选择器显示

如果在后期维护中，需要调整段落颜色，只需要修改 color 属性值即可。



CSS 语言对于所有属性和值都有相对严格要求，如果声明的属性没有在 CSS 规范中，或者某个属性值不符合属性要求，都不能使 CSS 语句生效。

11.1.2　类选择器

在一个页面中，使用标签选择器，会控制该页面中所有此类标记的显示样式。如果需要重新设定此类标记中的一个标记，仅使用标签选择器是不能达到效果的，还需要使用类（class）选择器。

类选择器用来为一系列标记定义相同的呈现方式，常用语法格式如下所示：

```
. classValue {property:value}
```

其中 classValue 是选择器的名称，具体名称由 CSS 制定者自己命名。如果一个标记具有 class 属性且 class 属性值为 classValue，那么该标记的呈现样式由该选择器指定。在定义类选择符时，需要在 classValue 前面加一个句点（.）。使用示例如下所示：

```
.rd{color:red}
.se{font-size:3px}
```

上面定义了两个类选择器，分别为 rd 和 se。类的名称可以是任意英文字符串或以英文开头的字符与数字的组合，一般情况下，类的名称是其功能及效果的简要缩写。

可以通过 p 标记的 class 属性来使用类选择符，用法如下所示：

```
<p class="rd">class 属性是用来引用类选择器的属性</p>
```

前面定义的选择器只能应用于指定的标记中（例如 p 标记）。如果需要在不同标记中使用相同的呈现方式，方法如下所示：

```
<p class="rd">段落样式</p>
<h3 class="rd">标题样式</h3>
```

【例 11.2】（实例文件：ch11\11.2.html）

```
<html>
<head><title>类选择器</title>
<style>
.aa{
    color:blue;
    font-size:20px;
}
.bb{
      color:red;
      font-size:22px;
}
</style></head><body>
<h3 class=bb>学习类选择器</h3>
<p class="aa">此处使用类选择器 aa 控制段落样式</p>
<p class="bb">此处使用类选择器 bb 控制段落样式</p>
</body>
</html>
```

在 IE 9.0 中浏览效果如图 11-2 所示，可以看到第一段以蓝色字体显示，大小为 20px，第二段以红色字体显示，大小为 22px，标题同样以红色字体显示，大小为 22px。

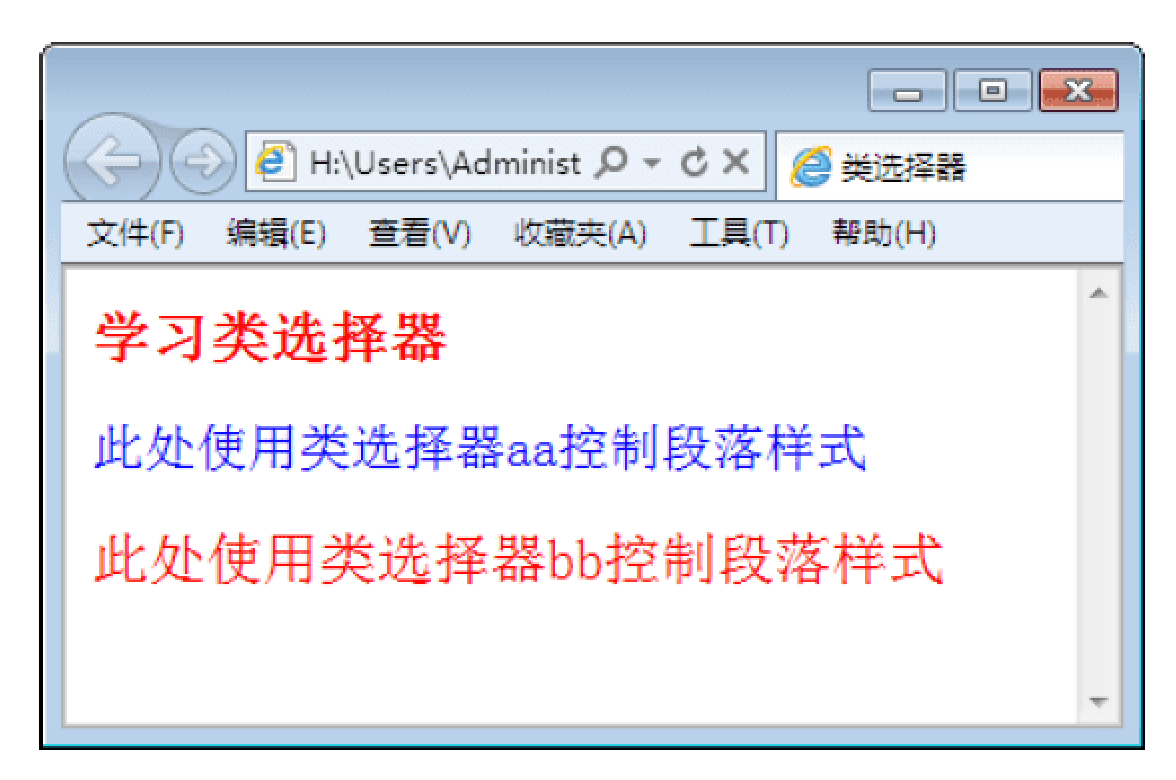


图 11-2　类选择器的显示

11.1.3　ID 选择器

ID 选择器和类选择器类似，都是针对特定属性的属性值进行匹配。ID 选择器定义的是某一个特定的 HTML 元素，一个网页文件中只能有一个元素使用某一 ID 的属性值。

定义 ID 选择器的基本语法格式如下所示：

```
#idValue{property:value}
```

在上述基本语法格式中，idValue 是选择器名称，可以由 CSS 定义者自己命名。如果某标记具有 id 属性，并且该属性值为 idValue，那么该标记的呈现样式由该 ID 选择器指定。在正常情况下 id 属性值在文档中具有唯一性。

下面定义了一个 ID 选择器：

```
#fontstyle
{
   color:red;
   font-weight:bold;
   font-size:large
}
```

在页面中，具有 ID 属性的标记才能够使用 ID 选择器定义样式，所以与类选择器相比，使用 ID 选择器具有一定的局限性。类选择器与 ID 选择器主要有两个区别。

- 类选择器可以给任意数量的标记定义样式，但 ID 选择器在页面的标记中只能使用一次。
- ID 选择器比类选择器具有更高的优先级，即当 ID 选择器与类选择器发生冲突时，优先使用 ID 选择器。

【例 11.3】（实例文件：ch11\11.3.html）

```
<html>
<head>
<title>ID 选择器</title>
```

```
<style>
#fontstyle{
     color:blue;
     font-weight:bold;
}
#textstyle{
       color:red;
       font-size:22px;
}
</style>
</head>
<body>
<h3 id=textstyle>学习 ID 选择器</h3>
<p id=textstyle>此处使用 ID 选择器 aa 控制段落样式</p>
<p id=fontstyle>此处使用 ID 选择器 bb 控制段落样式</p>
</body>
</html>
```

在 IE 9.0 中浏览效果如图 11-3 所示，可以看到第一个段落以红色字体显示，大小为 22px，第二段落以红色字体显示，字体加粗，标题同样以红色字体显示，大小为 20px。

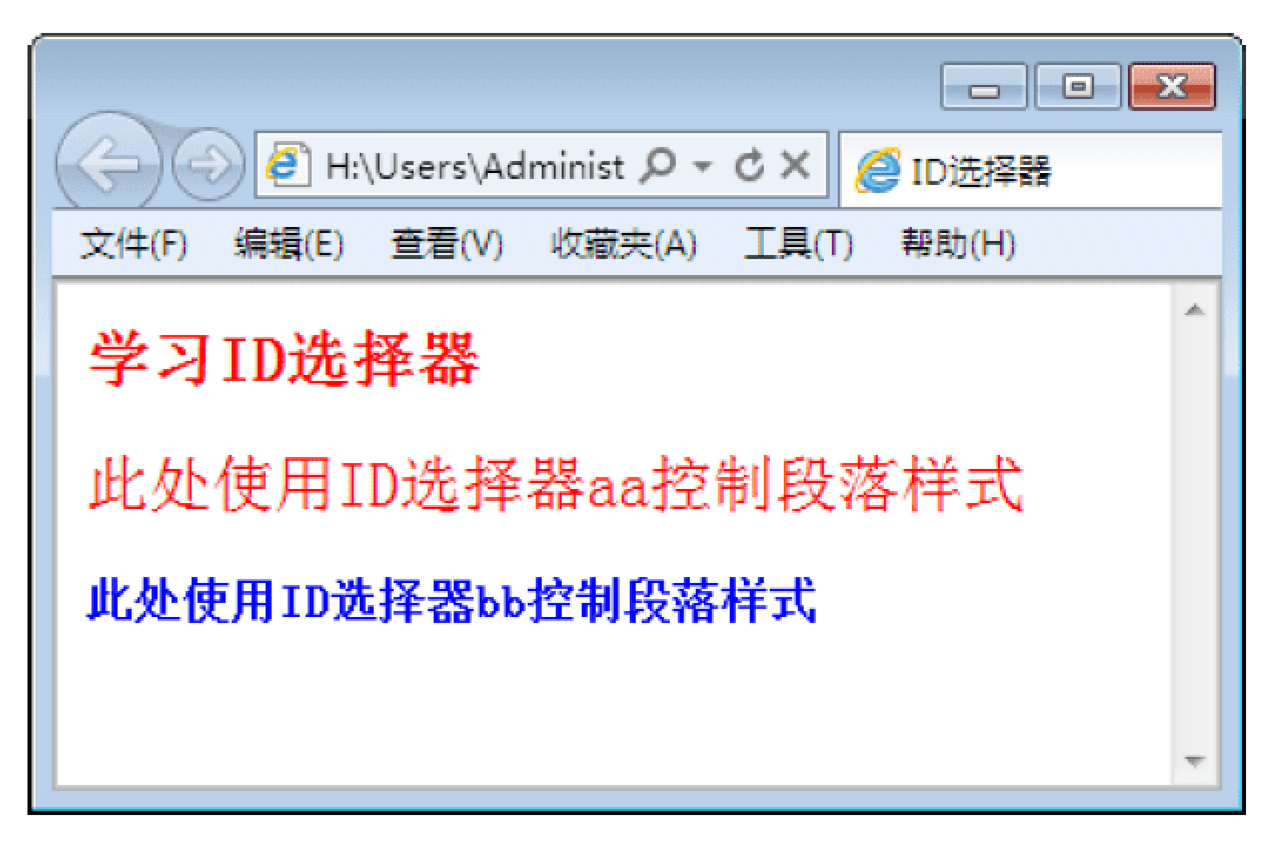


图 11-3　ID 选择器显示

从上面的代码可以看出，标题 h3 和第一个段落都使用了名称为 textstyle 的 ID 选择器，并都显示了 CSS 方案，在很多浏览器下，ID 选择器可以用于多个标记。但这里需要指出的是，将 ID 选择器用于多个标记是错误的，因为每个标记定义的 ID 不只有 CSS 可以调用，JavaScript 等脚本语言同样也可以调用。

11.1.4　全局选择器

如果想要让一个页面中的所有 html 标记使用同一种样式，可以使用全局选择器。顾名思义，全局选择器就是对所有 HTML 元素起作用。其语法格式为：

```
*{property:value}
```

其中“*”表示对所有元素起作用，property 表示 CSS 属性名称，value 表示属性值。用法如下所示：

```
*{margin:0; padding:0;}
```

【例 11.4】（实例文件：ch11\11.4.html）

```
<html>
<head><title>全局选择器</title>
<style>
*{
    color:red;
    font-size:30px
}
</style></head>
<body>
<p>使用全局选择器修饰</p>
<p>十年生死两茫茫，</p>
<h1>不思量，自难忘。</h1>
</body>
</html>
```

在 IE 9.0 中浏览效果如图 11-4 所示，可以看到两个段落和标题都以红色字体显示，大小为 30px。

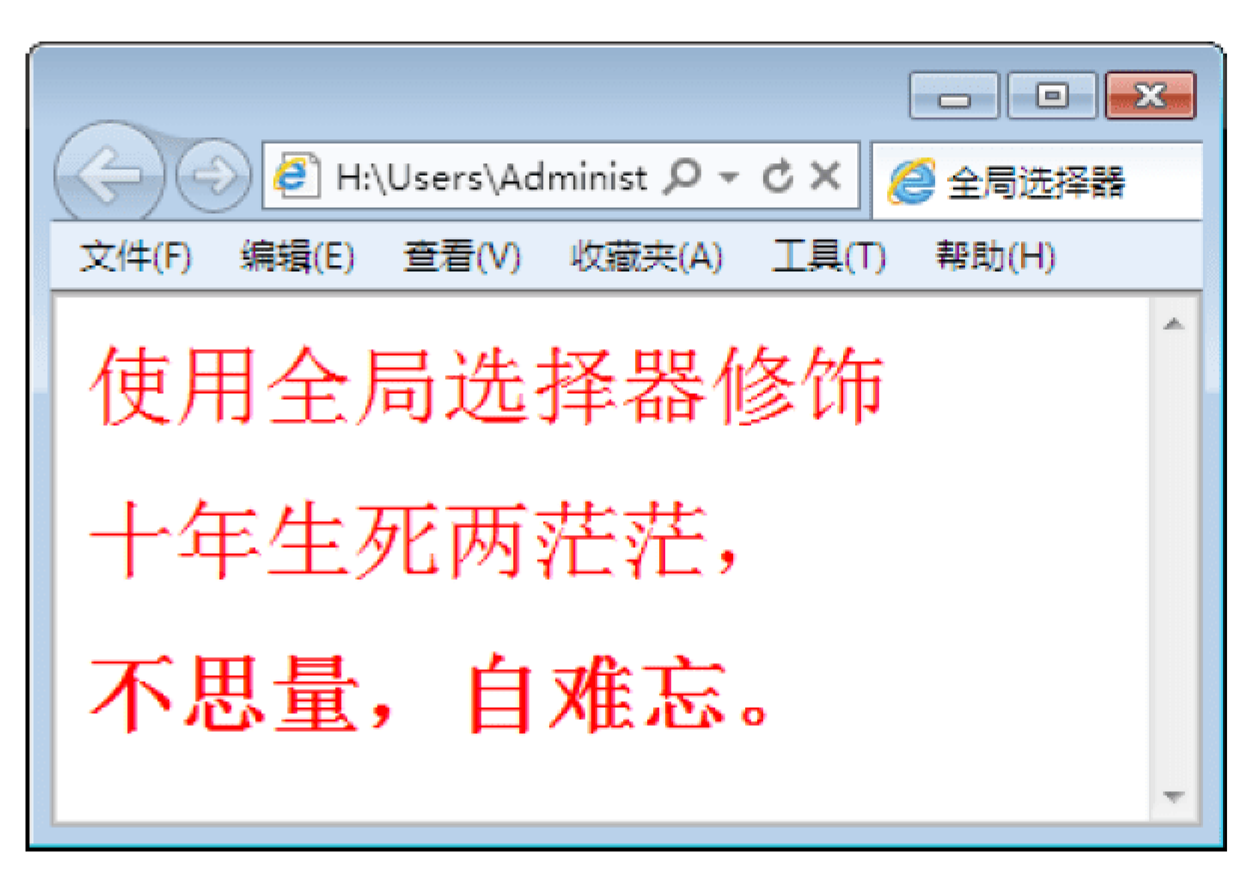


图 11-4 全局选择器显示

11.1.5 组合选择器

将多种选择器组合使用，可以构成一种复合选择器，也称为组合选择器。一般的组合方式是标签选择器和类选择器组合或标签选择器和 ID 选择器组合。由于这两种组合方式的原理和

效果一样，所以这里只介绍标签选择器和类选择器的组合。

组合选择器只是一种组合形式，不是一种真正的选择器，但在实际中经常使用。其用法如下所示：

```
.orderlist li {xxxx}
.tableset td {}
```

组合选择器一般用于重复出现并且样式相同的一些标签里，如 li 列表、td 单元格和 dd 自定义列表等。例如：

```
h1.red {color: red}
<h1 class="red"></h1>
```

【例 11.5】（实例文件：ch11\11.5.html）

```
<html>
<head>
<title>组合选择器</title>
<style>
p{
   color:red
}
p .firstPar{
   color:blue
}
.firstPar{
   color:green
}
</style></head><body>
<p>这是普通段落</p>
<p class="firstPar">此处使用组合选择器</p>
<h1 class="firstPar">我是一个标题</h1>
</body>
</html>
```

在 IE 9.0 中浏览效果如图 11-5 所示，可以看到第一个段落颜色为红色，采用的 p 标签选择器，第二个段显示的是蓝色，采用的 p 标签选择器和类选择器二者组合的选择器，标题 h1 以绿色字体显示，采用的是类选择器。

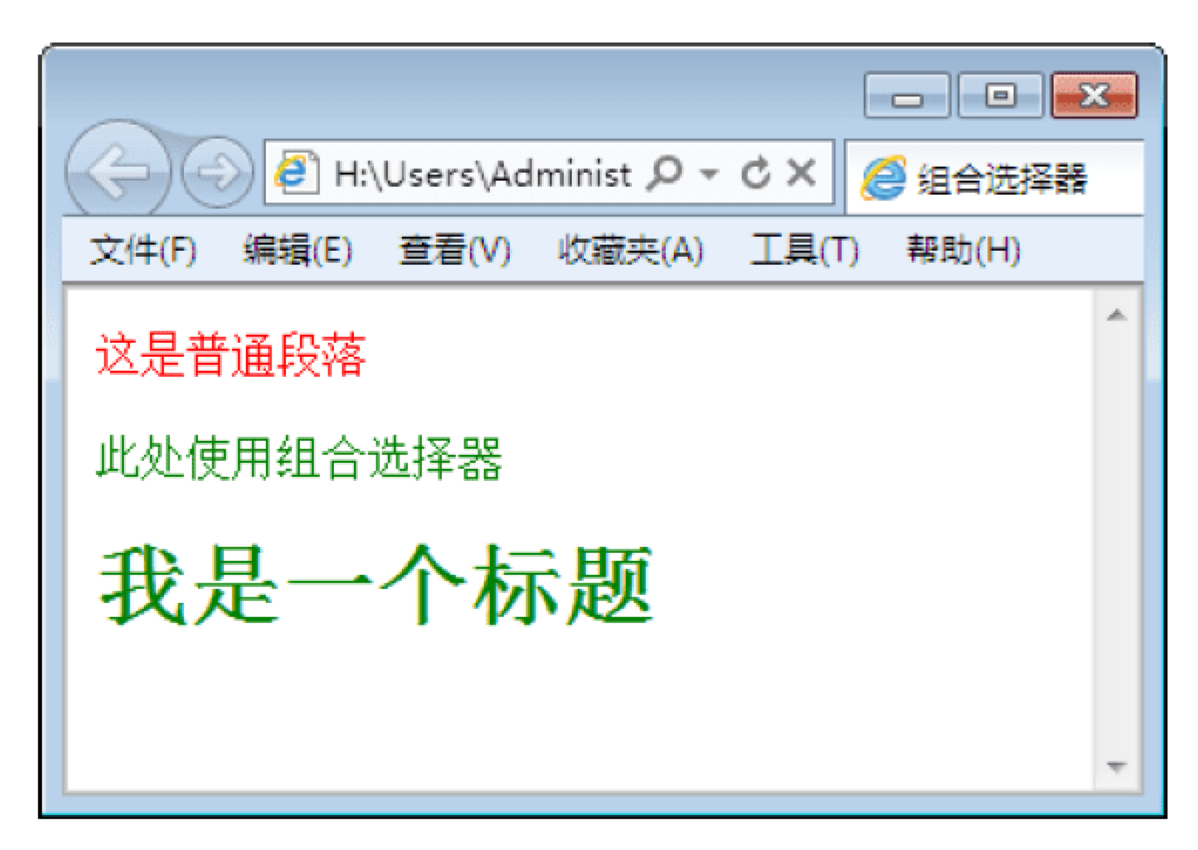


图 11-5　组合选择器显示

11.1.6　继承选择器

继承选择器的规则是，子标记在没有定义的情况下所有的样式是继承父标记的，当子标记重复定义了父标记已经定义过的声明时，子标记就执行后面的声明；与父标记不冲突的地方仍然沿用父标记的声明；

使用继承选择器就必须先了解 HTML 文档树和 CSS 继承，这样才能够较好地运用继承选择器。每个 HTML 都可以被看作一个文档树，文档树的根部就是 html 标记，而 head 和 body 标记就是其子元素。head 和 body 中的其他标记就是 html 标记的子孙元素。整个 HTML 就呈现一种祖先和子孙的树状关系。CSS 的继承是指子孙元素继承祖先元素的某些属性。

用法示例如下所示：

```
<div class="test">
<span><img src="xxx" alt="示例图片"/></span>
</div>
```

对于上面而言，如果其修饰样式为下面代码：

```
.test span img {border:1px blue solid;}
```

表示该选择器先找到 class 为 test 的标记，再从其子标记中查找 span 标记，然后从 span 的子标记中找到 img 标记。也可以采用下面的形式：

```
div span img {border:1px blue solid;}
```

可以看出其规律是从左往右，依次细化，最后锁定要控制的标记。

【例 11.6】（实例文件：ch11\11.6.html）

```
<html>
<head>
<title>继承选择器</title>
<style type="text/css">
```

```
h1{color:red; text-decoration:underline;}
h1 strong{color:#004400; font-size:40px;}
</style>
</head>
<body>
<h1>测试 CSS 的<strong>继承</strong>效果</h1>
<h1>此处使用继承<font>选择器</font>了么？</h1>
</body>
</html>
```

在 IE 9.0 中浏览效果如图 11-6 所示，可以看到第一个段落颜色为红色，但是“继承”两个字使用绿色显示，并且大小为 40px，其他的 CSS 样式都是继承父标记<h1>的样式，例如下划线设置。第二个标题中，虽然使用了 font 标记修饰“选择器”，但其样式都继承于父类标记<h1>。

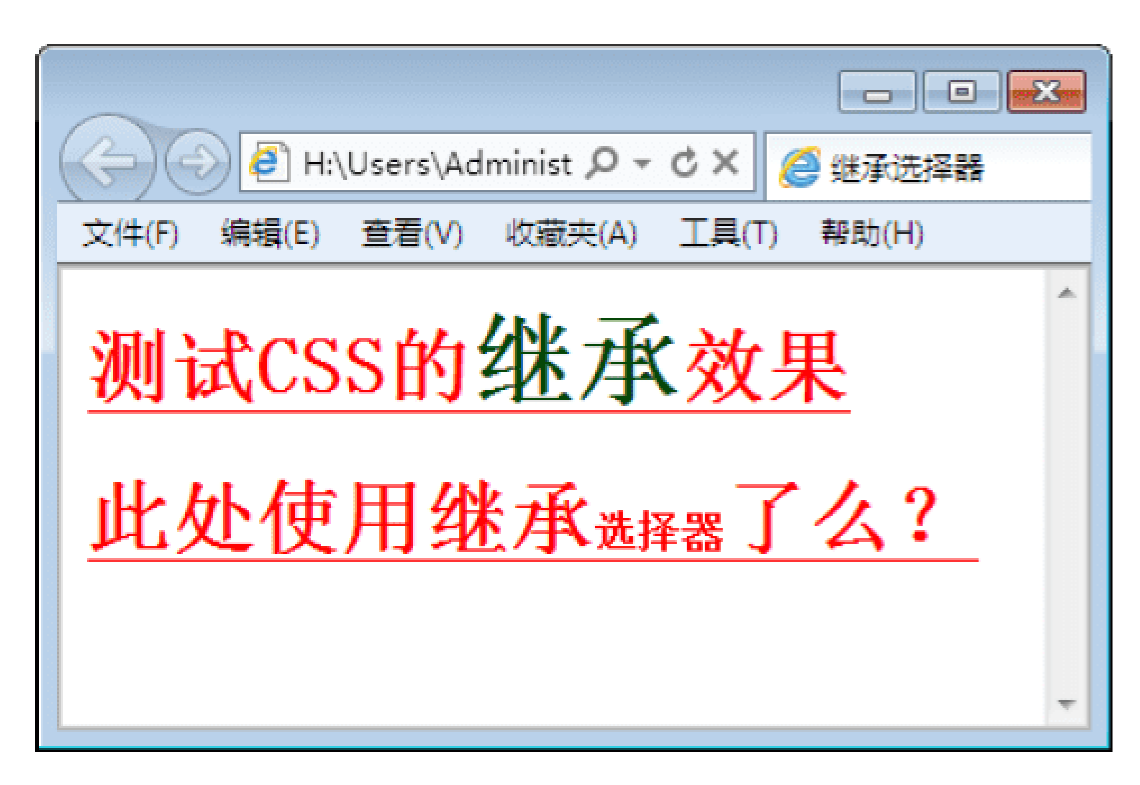


图 11-6　继承选择器显示

11.1.7　伪类选择器

伪类也是选择器的一种，但是用伪类定义的 CSS 样式并不是作用在标记上的。伪类作用在标记的状态上。伪类包括 :first-child、:link:、:vistited、:hover、:active、:focus 和:lang 等。其中有一组伪类是主流浏览器都支持的，就是超链接的伪类，包括:link:、:vistited、:hover 和:active。



由于很多浏览器支持不同类型的伪类，没有统一的标准，所以很多伪类通常不会用到。

伪类选择符定义的样式最常用于标记<a>上，它表示链接 4 种不同的状态：未访问链接（link）、已访问链接（visited）、激活链接（active）和鼠标停留在链接上（hover）。

要注意的是，<a>可以只具有一种状态（:link），或者同时具有两种或三种状态。例如，任何一个有 HREF 属性的<a>标签，在未有任何操作时都已经具备了:link 的条件，也就是满足了有链接属性这个条件；访问过的<a>标记，同时会具备 :link 和:visited 两种状态。把鼠标移

到访问过的<a>标记上的时候，<a>标记就同时具备了 :link、:visited 和:hover 三种状态。用法示例如下所示：

```
a:link{color:#FF0000; text-decoration:none}
a:visited{color:#00FF00; text-decoration:none}
a:hover{color:#0000FF; text-decoration:underline}
a:active{color:#FF00FF; text-decoration:underline}
```

上面的样式表示该链接未访问时颜色为红色且无下划线，访问后是绿色且无下划线，激活链接时为蓝色且有下划线，鼠标放在链接上为紫色且有下划线。

【例 11.7】（实例文件：ch11\11.7.html）

```
<html>
<head>
<title>伪类</title>
<style>
a:link {color: red}             /* 未访问的链接 */
a:visited {color: green}            /* 已访问的链接 */
a:hover {color:blue}            /* 鼠标移动到链接上 */
a:active {color: orange}            /* 选定的链接 */
</style>
</head>
<body>
<a href="">链接到本页</a>
<a href="http://www.sohu.com">搜狐</a>
</body>
</html>
```

在 IE 9.0 中浏览效果如图 11-7 所示，可以看到两个超级链接，第一个是鼠标停留在上方时，超级链接显示颜色为蓝色，另一个是访问过后，超级链接显示颜色为绿色。

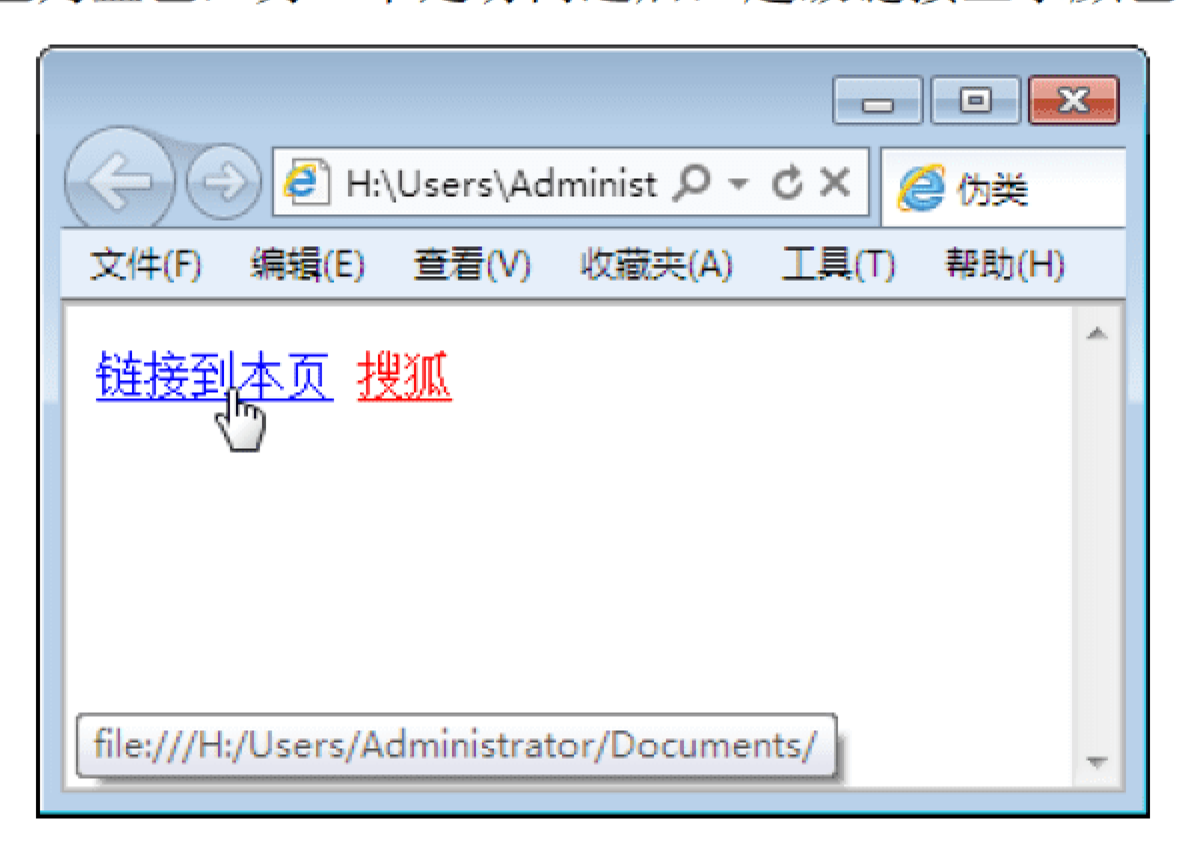


图 11-7　伪类显示

11.1.8 属性选择器

前面在使用 CSS 样式对 HTML 标记进行修饰时，都是通过 HTML 标记名称或自定义名称指向具体 HTML 元素的，进而控制 HTML 标记样式。那么能不能直接通过标记属性，而不是标记名称或自定义名称来进行修饰呢？直接使用属性控制 HTML 标记样式，称为属性选择器。

属性选择器就是根据某个属性是否存在或属性值来寻找元素，因此能够获得有趣和强大的效果。

【例 11.8】（实例文件：ch11\11.8.html）

```
<html>
<head>
<title>属性选择器</title>
<style>
[align]{color:red}
[align="left"]{font-size:20px;font-weight:bolder;}
[lang^="en"]{color:blue;text-decoration:underline;}
[src$="gif"]{border-width:5px;boder-color:#ff9900}
</style>
</head>
<body>
<p align=center>这是使用属性定义样式</p>
<p align=left>这是使用属性值定义样式</p>
<p lang="en-us">此处使用属性值前缀定义样式</p>
<p>下面使用了属性值后缀定义样式</p>
<img src="2.gif" border="1"/>
</body>
</html>
```

在 Firefox 5.0 中浏览效果如图 11-8 示，可以看到第一个段落使用属性 align 定义样式，其字体颜色为红色。第二个段落使用属性值 left 修饰样式，并且大小为 20px，加粗显示，其字体颜色为红色，是因为该段落使用了 align 这个属性。第三个段落显示红色，且带有下划线，是因为属性 lang 的值前缀为 en。最后的图片以边框样式显示，是因为属性值后缀为 gif。

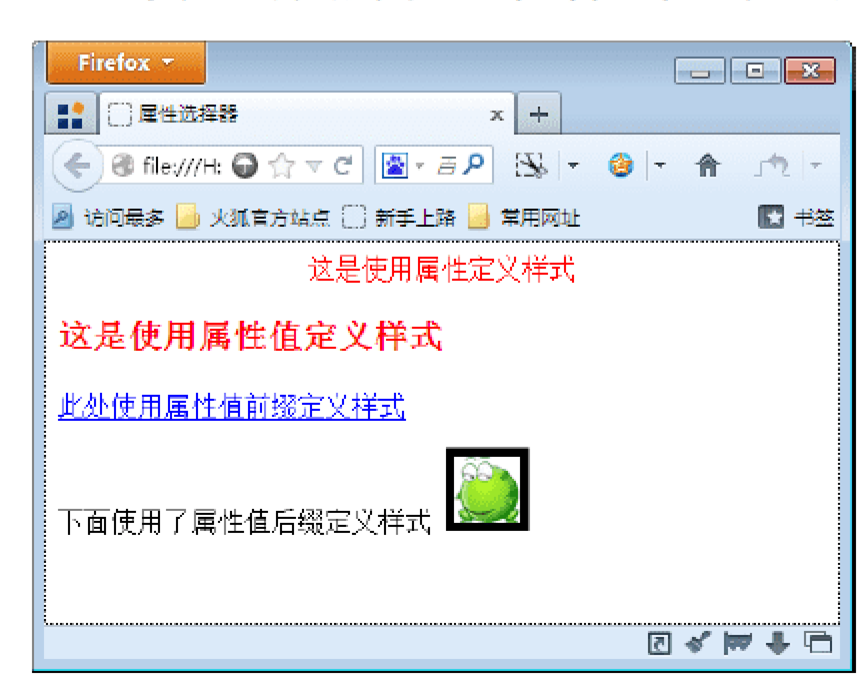


图 11-8 属性选择器显示

11.2 选择器声明

使用 CSS 选择器可以控制 HTML 标记样式，其中每个选择器属性可以一次声明多个，即创建多个 CSS 属性修饰 HTML 标记，实际上也可以将选择器声明多个，并且任何形式的选择器都是合法的。

11.2.1 集体声明

在一个页面中，有时需要将不同种类的标记样式保持一致，例如需要 p 标记和 h1 字体保持一致，此时可以使 p 标记和 h1 标记共同使用类选择器，除这个方法之外，还可以使用集体声明方法。集体声明就是在声明各种 CSS 选择器时，如果某些选择器的风格完全相同，或者部分相同，可以将风格相同的 CSS 选择器同时声明。

【例 11.9】（实例文件：ch11\11.9.html）

```
<html>
<head>
<title>集体声明</title>
<style type="text/css">
 h1,h2, h3,p{
      color:red;
      font-size:20px;
      font-weight:bolder;
}
</style></head><body>
<h1>让我怎么感谢您，</h1>
<h2>当我走向你的时候，</h2>
<h3>我原想收获一缕春风，</h3>
<p>你却给了我整个春天。</p>
</body>
</html>
```

在 IE 9.0 中浏览效果如图 11-9 所示，可以看到网页中的标题 1、标题 2、标题 3 和段落都以红色字体加粗显示，并且大小都为 20px。

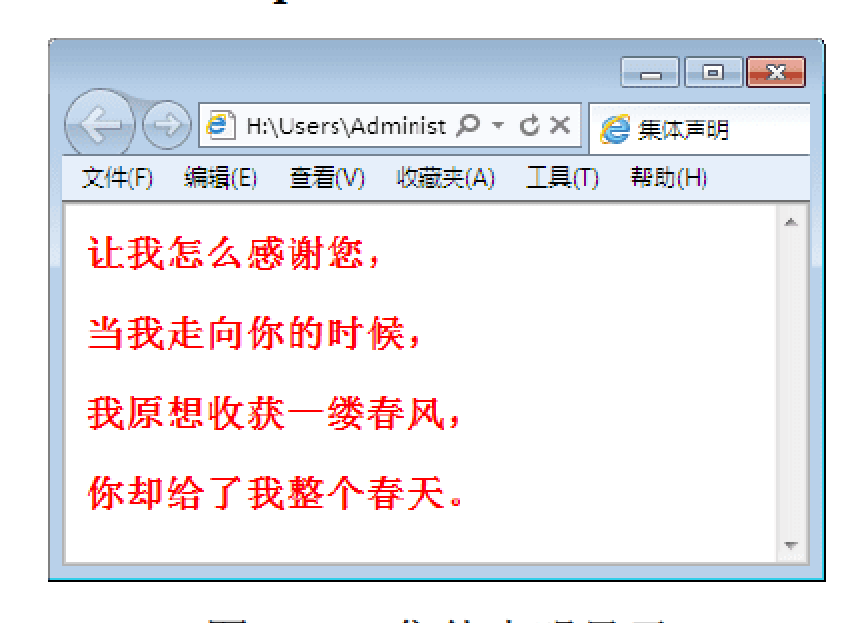


图 11-9 集体声明显示

11.2.2 多重嵌套声明

在通过 CSS 控制 HTML 标记样式时，还可以使用层层递进的方式，即嵌套方式，对指定位置的 HTML 标记进行修饰，例如当<p>与</p>标记之间包含<a>标记时，就可以使用这种方式对 HMTL 标记进行修饰。

【例 11.10】（实例文件：ch11\11.10.html）

```
<html>
<head>
<title>多重嵌套声明</title>
<style>
p{font-size:20px;}
p a{color:red;font-size:30px;font-weight:bolder;}
</style></head><body>
<p>这是一个多重嵌套<a href="">测试</a></p>
</body>
</html>
```

在 IE 9.0 中浏览效果如图 11-10 所示，可以看到在段落中，超级链接显示为红色字体，大小为 30px，其原因是使用嵌套声明。

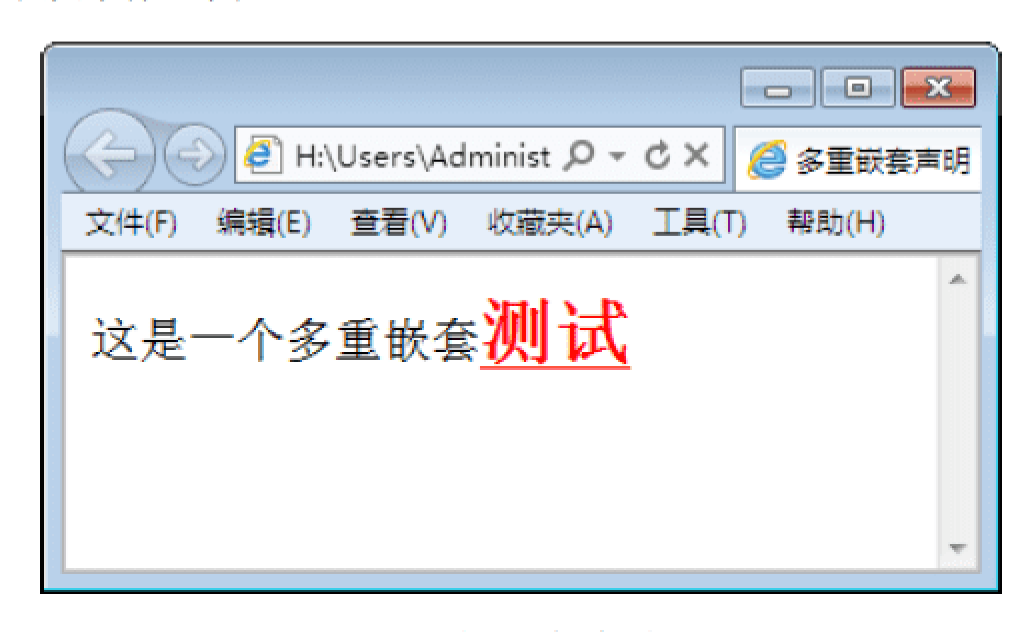


图 11-10 多重嵌套声明显示

11.3 CSS 继承

可以将 HTML 网页看作是一个节点的集合，在一个 HTML 文档中可以包含不同的标记，HTML 文档中的每个成份都是一个节点。一个节点树中的所有节点彼此都是有关系的。节点树可以把一个 HTML 文档展示为一个节点集，以及它们之间的连接。在一个节点树中，最顶端的节点被称为根。除根之外，每一个节点都拥有父节点。一个节点可以有无限的子，叶是无子的节点，同级节点指拥有相同的父的节点。HTML 树结构如图 11-11 所示。

CSS 继承指的是子标记会继承父标记的所有样式风格，并可以在父标记样式风格的基础上加以修改，产生新的样式，而子标记样式风格完全不会影响父标记。

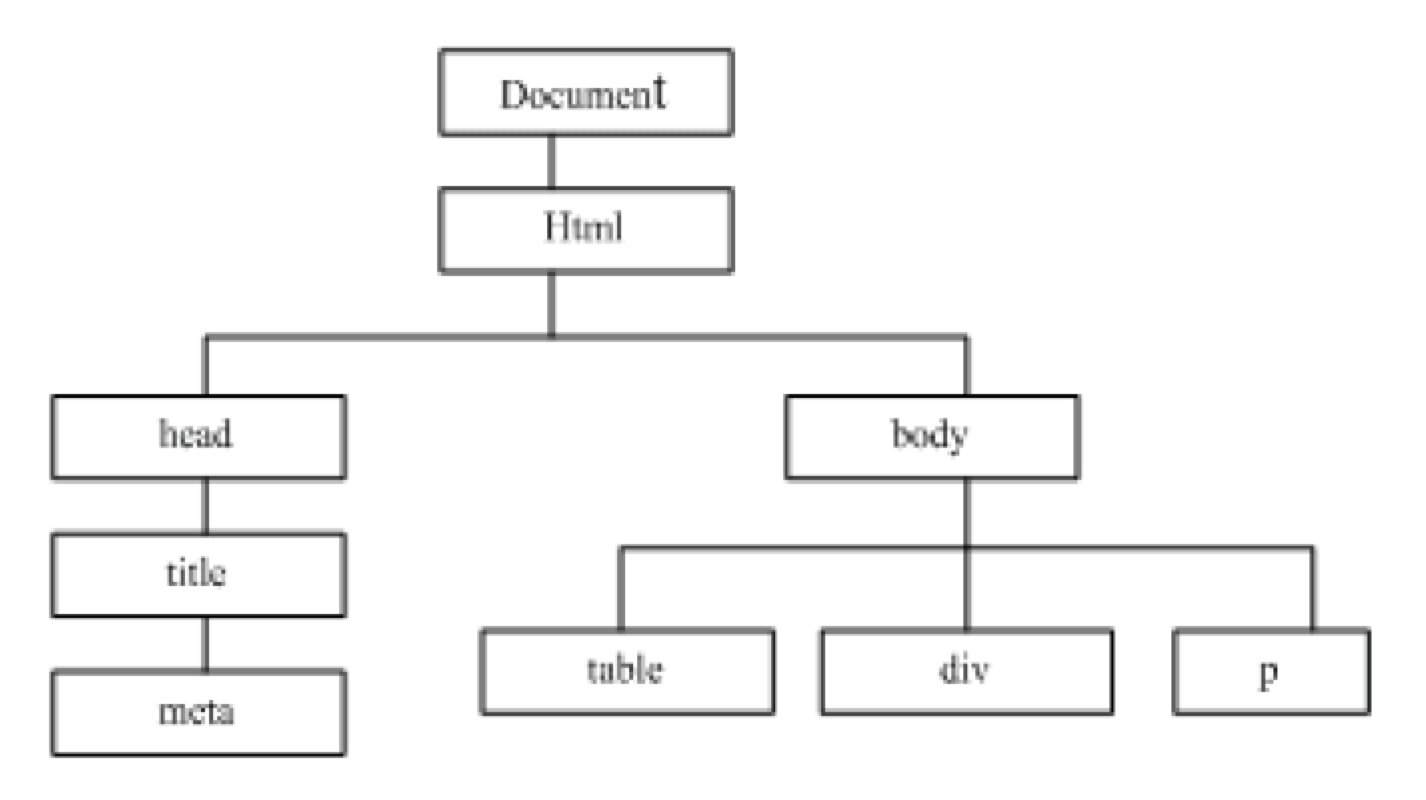


图 11-11　HTML 树结构

【例 11.11】（实例文件：ch11\11.11.html）

```
<html>
<head><title>多重嵌套声明</title>
<style>
p{font-size:20px;color:red}
span{font-size:30px;}
</style></head><body>
<p>这是一个继承<span>测试</span></p>
</body>
</html>
```

在 IE 9.0 中浏览效果如图 11-12 所示，可以看到，在段落中字体颜色为红色，大小为 20px，但段落 span 标记中的文本为红色字体，大小为 30px，此样式首先继承了父标记中的颜色样式，并重新定义了自己的样式。

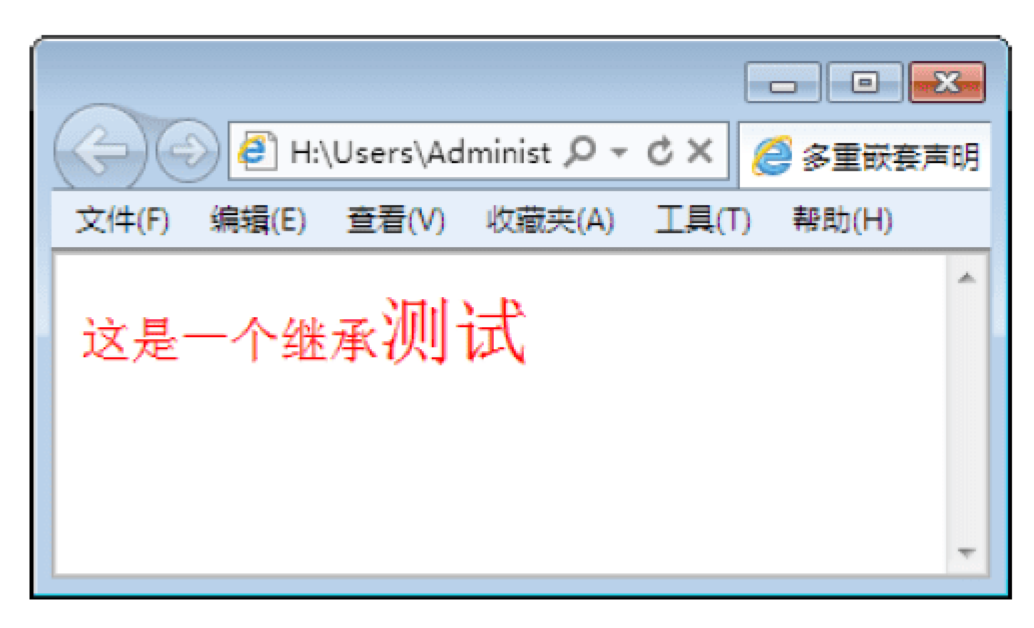


图 11-12　CSS 继承显示



CSS 继承贯穿整个 CSS 设计的始终，每个标记都遵循着 CSS 继承的概念。同样可以利用这种巧妙的继承关系，大大缩减代码的编写量，并提高可读性，尤其是在页面内容很多和父子关系庞杂的时候。

11.5　综合实例——制作新闻菜单

在网上浏览新闻是每个上网者都喜欢做的事情。一个布局合理、样式美观的新闻菜单是吸引人的主要途径之一。本实例使用 CSS 控制 HTML 标记，创建新闻菜单。具体步骤如下所示。

01 分析需求。

创建一个新闻菜单，需要包含两个部分，一个是父菜单，用来表明新闻类别，一个是子菜单，介绍具体的新闻消息。创建菜单的方式有很多，可以用 table 创建，也可以用列表创建，还可以使用段落 p 创建。本实例采用 p 标记结合 div 创建。完成后的效果如图 11-13 所示。

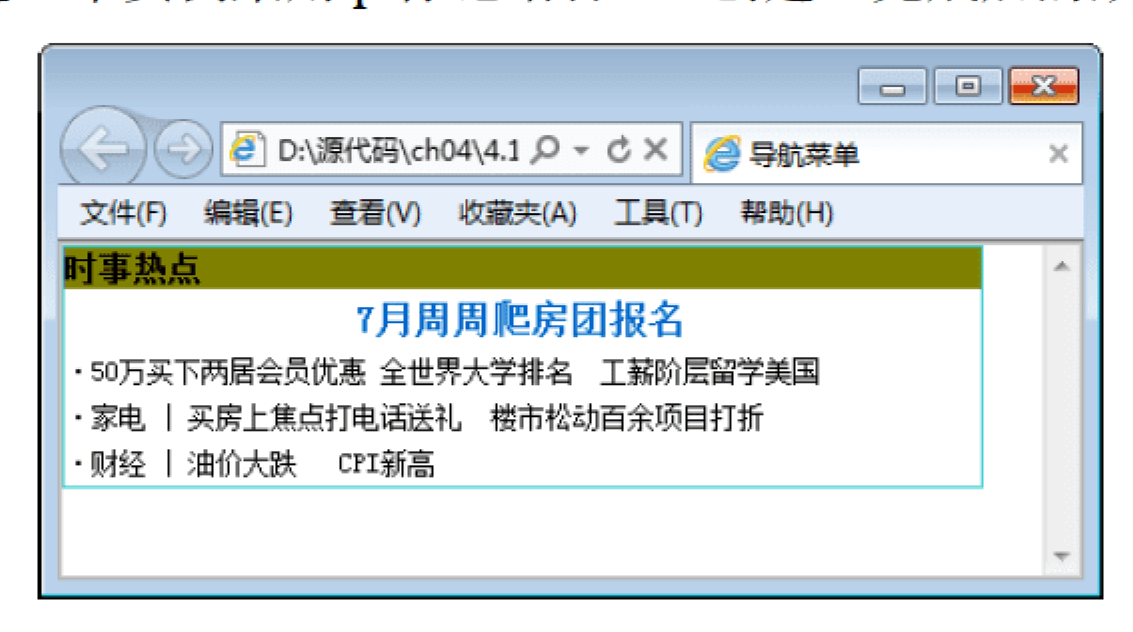


图 11-13　新闻菜单显示

02 分析局部和整体，构建 HTML 网页。

在一个新闻菜单中，可以分为 3 个层次，一个新闻父菜单，一个新闻焦点，一个新闻子菜单，分别使用 div 创建。其 HTML 代码如下所示。

```
<html >
<head><title>导航菜单</title>
</head><body>
        <div class="big">
        <h2>时事热点 </h2>
        <div class="up">
                    <a href="#">7 月周周爬房团报名</a>
              </div> <div class="down">
                <p> • 50 万买下两居会员优惠  全世界大学排名  工薪阶层留学美国</p><p>
                     • 家电 | 买房上焦点打电话送礼  楼市松动百余项目打折</p><p>
                     • 财经 | 油价大跌   CPI 新高  </p>
              </div> </div>
</body>
</html>
```

在 IE 9.0 中的游览效果，如图 11-14 所示。可以看到一个标题、一个超级链接和 3 个段落，以普通样式显示。

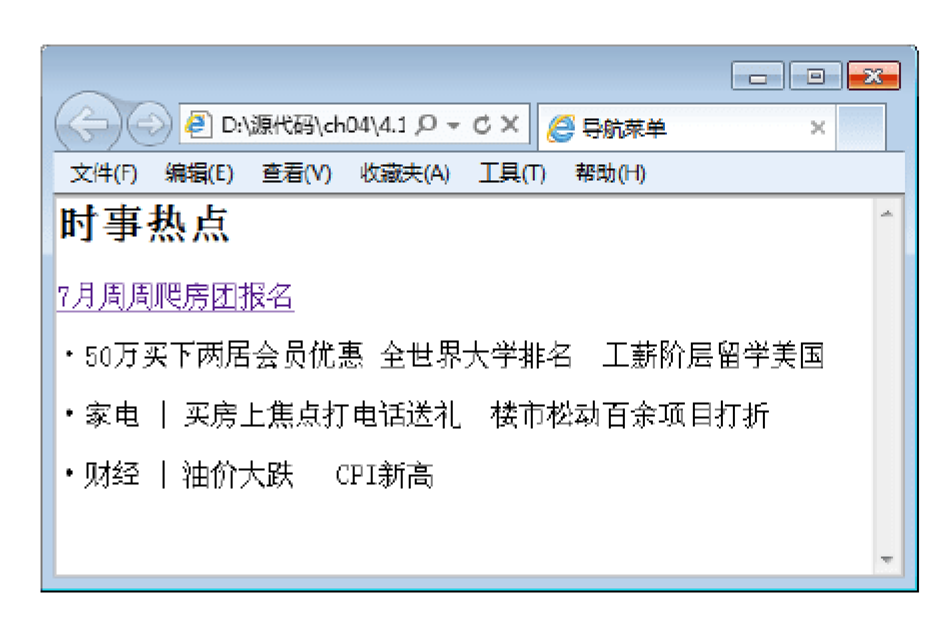


图 11-14　无 CSS 标记显示

03 添加 CSS 代码，修饰整体样式。

对于 HTML 页面，需要有一个整体样式，其代码如下所示：

```
*{
    padding:0px;
    margin:0px;
}
body{
    font-family:"宋体";
    font-size:12px;
    }
.big{
    width:400px;
    border:#33CCCC 1px solid;
}
```

在 IE 9.0 中的浏览效果如图 11-15 所示。可以看到全局层 div 会以边框显示，宽度为 400 像素，其颜色为浅绿色，body 文档内容中字体采用宋体，大小为 12 像素，并且定义内容和层之间空隙为零，层和层之间空隙为零。

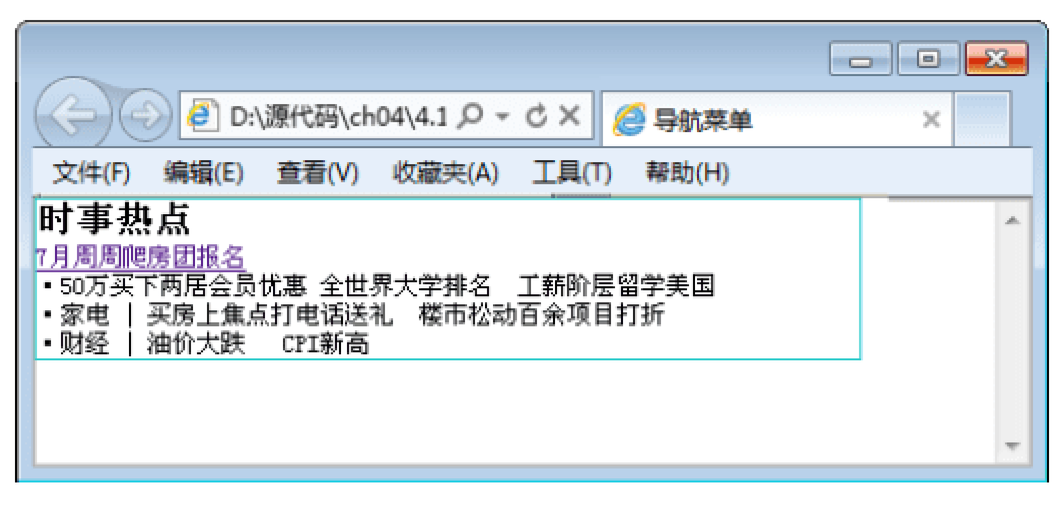


图 11-15　整体样式添加

04 添加 CSS 代码，修饰新闻父菜单。

对新闻父类菜单进行 CSS 控制，其代码如下所示：

```
h2{background-color:olive;
      display:block;
```

```
        width:400px;
        height:18px;
        line-height:18px;
        font-size:14px;}
```

在 IE 9.0 中的游览效果如图 11-16 所示。可以看到超级链接“时事热点”会以矩形方框显示，其背景色为橄榄色，字体大小为 14 像素，行高为 18 像素。

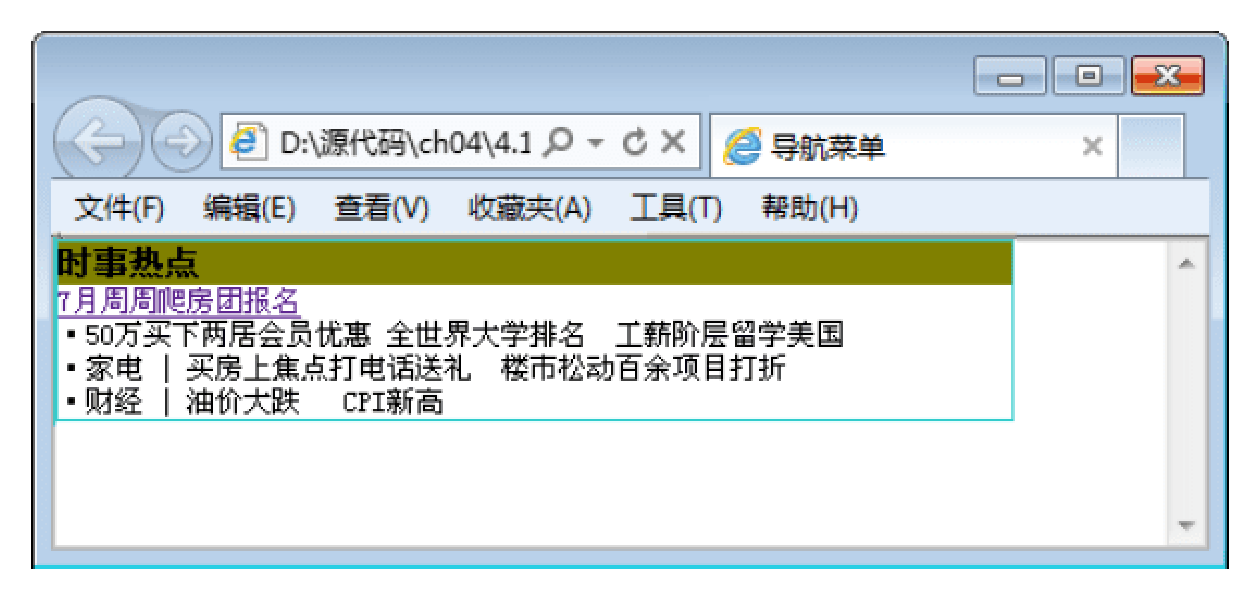


图 11-16 修饰超级链接

05 添加 CSS 菜单，修饰子菜单。

```
.up{padding-bottom:5px;
        text-align:center; }
p{line-height:20px;}
```

在 IE 9.0 中的游览效果如图 11-17 所示。可以看到“7 月周周爬房团报名”居中显示，即在第二层 div 中使用类标记 up 修饰。所有段落之间间隙增大，即为 p 标记设置行高。

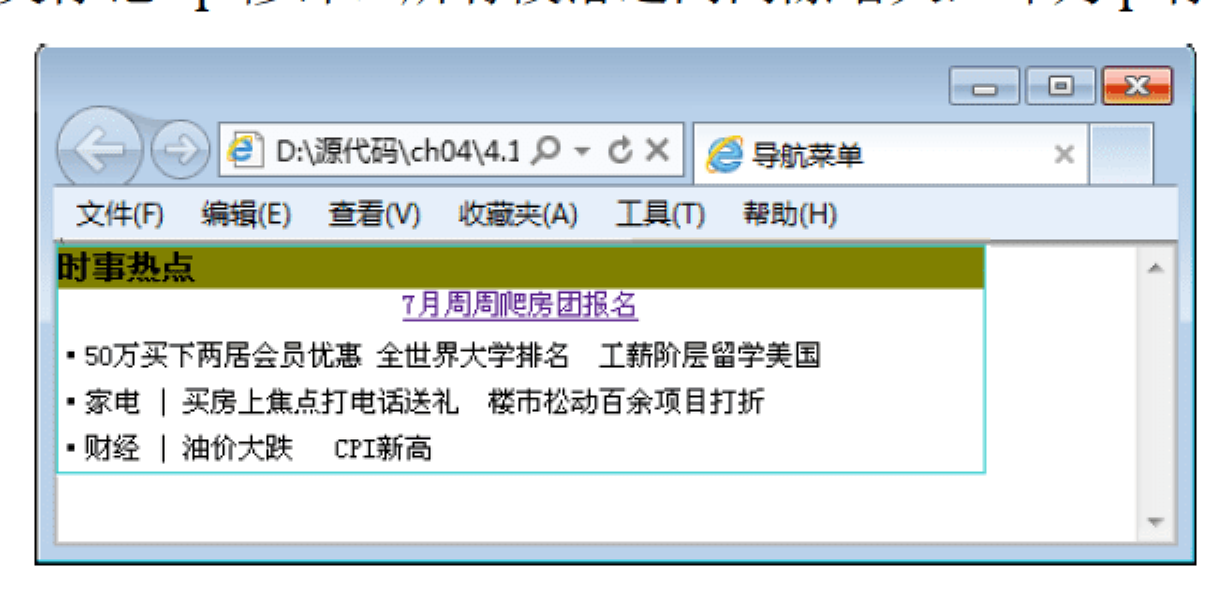


图 11-17 子菜单样式显示

06 添加 CSS 菜单，修饰超级链接。

```
a{font-size:16px;
        font-weight:800;
        text-decoration:none;
        margin-top:5px;
        display:block;}
a:hover{color:#FF0000;
        text-decoration:underline;}
```

在 IE 9.0 中的游览效果如图 11-18 所示。可以看到“7 月周周爬房团报名”字体变大，并且加粗，无下划线显示，当鼠标放在此超级链接上时，以红色字体显示，并且下面带有下划线。

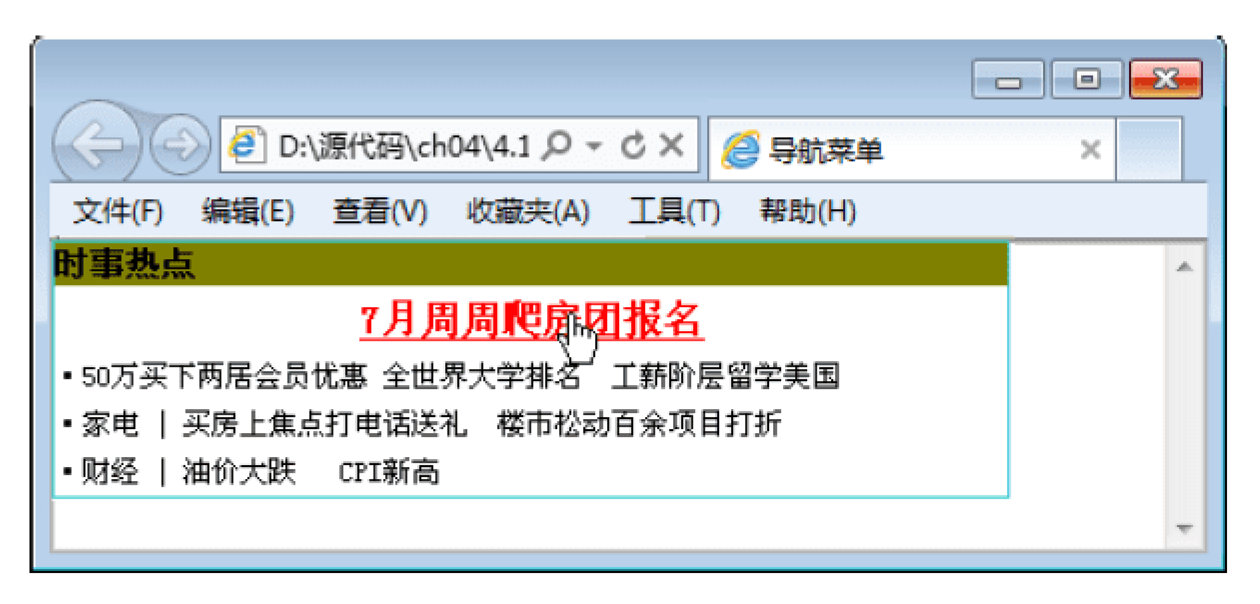


图 11-18 超级链接修饰显示

11.6 专家解惑

1. CSS+DIV 布局有什么好处？

使用 CSS+DIV 布局有下面几个好处：

（1）大大缩减页面代码，提高页面浏览速度，减少带宽成本。

（2）结构清晰，容易被搜索引擎搜索到。

（3）缩短改版时间。只要简单地修改几个 CSS 文件就可以重新设计一个具有成百上千个页面的站点。

（4）强大的字体控制和排版能力。CSS 控制字体的能力比糟糕的 FONT 标签好多了，有了 CSS，我们不再需要用 FONT 标签或者透明的 1 px GIF 图片来控制标题，或改变字体颜色、字体样式了。

2. 如何下载网页中的 CSS 文件？

选择网页上面的【查看】➢【源文件】菜单命令，如果有 CSS，可以直接复制下来，如果没有，可以找找有没有类似于这种连接的代码，例如：

```
<link href="/index.css" rel="stylesheet" type="text/css">
```

在打开的网址后面直接加“/index.css”，然后回车就可以打开 CSS 文件了。

3. 学了 JavaScript 就不用学 CSS 了吗？

JS 主要应用于动态表现（注意不是交互，交互使用的 AJAX 技术还是含有 JS 的），而 CSS 用于静态表现。JS 的动态内容是 CSS 永远不能做到的。但 CSS 不是一个辅助，它是网页表现的灵魂。

第 12 章

CSS字体与段落属性

常见的网站、博客是使用文字或图片来阐述自己观点，其中文字是传递信息的主要手段。而美观大方的网站或者博客，需要使用 CSS 样式修饰。设置文本样式是 CSS 技术的基本使命，通过 CSS 文本标记语言，可以设置文本的样式和粗细等。

12.1 CSS 文字样式

一个杂乱无序的网页，会使人产生枯燥无味的感觉。而一个美观大方的网页，会让人有流连忘返的感觉。可以通过 CSS 字体样式来设置这种效果。通过本节的 CSS 学习，会使读者设计出令人难忘的网页。

12.1.1 字体（font-family）

font-family 属性用于指定字体类型，如宋体、黑体、隶书、Times New Roman 等，即在网页中展示字体不同的形状。具体的语法如下所示。

```
{font-family : name}
{font-family : cursive | fantasy | monospace | serif | sans-serif}
```

从语法格式上可以看出，font-family 有两种声明方式。第一种方式，使用 name 字体名称按优先顺序排列，以逗号隔开，如果字体名称包含空格，则应使用引号括起，在 CSS 中，比较常用的是第一种声明方式。第二种声明方式使用所列出的字体序列名称。如果使用 fantasy 序列，将提供默认字体序列。

【例 12.1】（实例文件：ch12\12.1.html）

```
<html>
<style type=text/css>
p{font-family:黑体}
</style>
<body>
<p align=center>天行健，君子以自强不息。</p>
</body>
</html>
```

在 IE 9.0 中的浏览效果如图 12-1 所示，可以看到文字居中并以黑体显示。

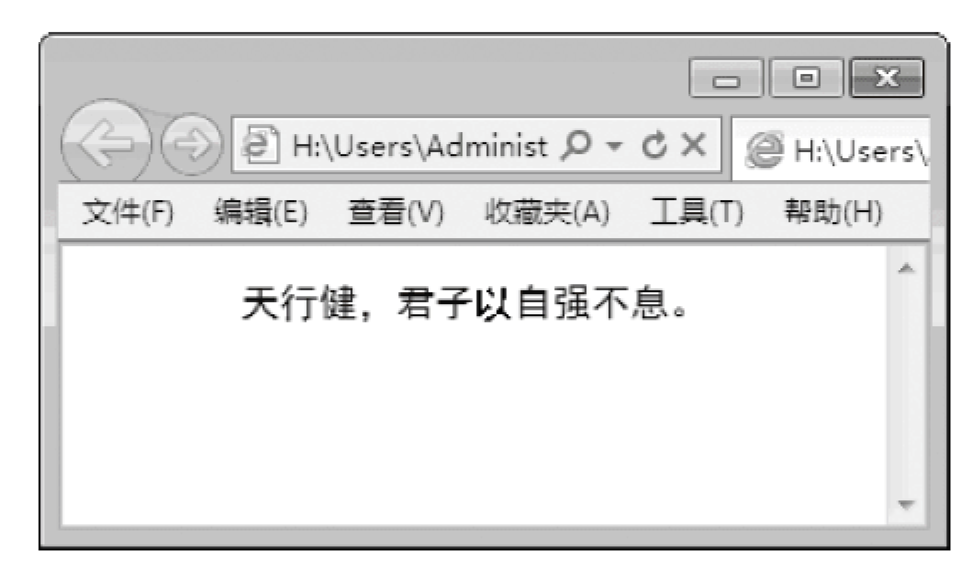


图 12-1 字型显示

在显示字体时，如果指定一种特殊字体类型，而在浏览器或者操作系统中该类型不能正确获取，可以通过 font-family 预设多种字体类型。font-family 属性可以预置多个供页面使用的字体类型，即字体类型序列，其中每种字型之间使用逗号隔开。如果前面的字体类型不能够正确显示，则系统将自动选择后一种字体类型，依次类推。

所以，在设计页面时，一定要考虑字体的显示问题，为了保证页面达到预计的效果，最好提供多种字体类型，而且最好以最基本的字体类型作为最后一个选择。

其样式设置如下所示：

```
p
{
  font-family:华文彩云,黑体,宋体
}
```

当 font-family 属性值中的字体类型由多个字符串和空格组成时，例如 Times New Roman，那么，该值就需要使用双引号引起来。

```
p
{
  font-family: "Times New Roman"
}
```

12.1.2 字号（font-size）

在网页中，标题通常使用较大字体显示，用于引人注意，小号字体用来显示正常内容，大小字体结合形成网页，既吸引了人们的眼球，又提高了阅读速度。

在 CSS 规定中，通常使用 font-size 设置文字大小。其语法格式如下所示。

```
{font-size : 数值| inherit | xx-small | x-small | small | medium | large | x-large | xx-large | larger | smaller | length}
```

其中，可以通过数值来定义字体大小，例如用 font-size:10px 的方式定义字体大小为 10 个像素。此外，还可以通过 medium 之类的参数定义字体的大小，其参数含义如表 12-1 所示。

表 12-1　font-size 参数列表

参数	说明
xx-small	绝对字体尺寸。根据对象字体进行调整。最小
x-small	绝对字体尺寸。根据对象字体进行调整。较小
small	绝对字体尺寸。根据对象字体进行调整。小
medium	默认值。绝对字体尺寸。根据对象字体进行调整。正常
large	绝对字体尺寸。根据对象字体进行调整。大
x-large	绝对字体尺寸。根据对象字体进行调整。较大
xx-large	绝对字体尺寸。根据对象字体进行调整。最大
larger	相对字体尺寸。相对于父对像中字体尺寸进行相对增大。使用成比例的 em 单位计算
smaller	相对字体尺寸。相对于父对像中字体尺寸进行相对减小。使用成比例的 em 单位计算
length	百分数或由浮点数字和单位标识符组成的长度值，不可为负值。其百分比取值基于父对象中字体的尺寸

【例 12.2】（实例文件：ch12\12.2.html）

```
<html>
<body>
<div style="font-size:10pt">旧梦依稀 往事迷离 春花秋月里
  <p style="font-size:small">如雾里看花 水中望月 漂来又浮去</p>
  <p style="font-size:larger">君来有声 君去无语 翻云覆雨里</p>
  <p style="font-size:x-small">虽两情相惜 两心相仪 得来复失去</p>
  <p style="font-size:x-larger">有诗待和 有歌待应 有心待相系</p>
  <p style="font-size:50%">望长相思 望长相守 却空留琴与笛</p>
  <p style="font-size:25pt">以情相悦 以心相许 以身相偎依</p>
</div>
</body>
</html>
```

在 IE 9.0 中的浏览效果如图 12-2 所示，可以看到网页中的文字被设置成不同的大小，其设置方式采用了绝对数值、关键字和百分比等形式。

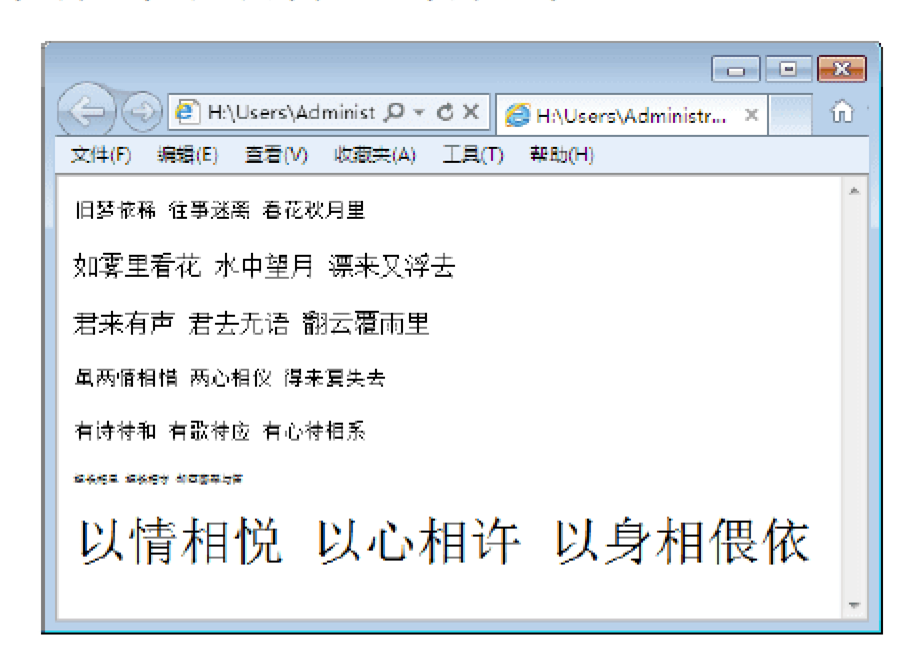


图 12-2　字体大小显示

在上面的例子中，font-size 字体大小为 50%时，其比较对象是上一级标记中的 10pt。同样我们还可以使用 inherit 值，直接继承上级标记的字体大小。例如：

```
<div style="font-size:50pt">上级标记
  <p style="font-size: inherit ">继承</p>
</div>
```

12.1.3　字体风格（font-style）

font-style 通常用来定义字体风格，即字体的显示样式。在 CSS 规定中，其语法格式如下所示。

```
font-style : normal | italic | oblique |inherit
```

其属性值有 4 个，具体含义如表 12-2 所示。

表 12-2　font-style 属性表

属性值	含义
Normal	默认值。浏览器显示一个标准的字体样式
italic	浏览器会显示一个斜体的字体样式
oblique	将没有斜体变量的特殊字体，浏览器会显示一个倾斜的字体样式
inherit	规定应该从父元素继承字体样式

【例 12.3】（实例文件：ch12\12.3.html）

```
<html>
<body>
  <p style="font-style:italic">梅花香自苦寒来</p>
  <p style="font-style:normal">梅花香自苦寒来</p>
  <p style="font-style:oblique">梅花香自苦寒来</p>
</body>
</html>
```

在 IE 9.0 中浏览效果如图 12-3 所示，可以看到文字分别显示不同的样式。

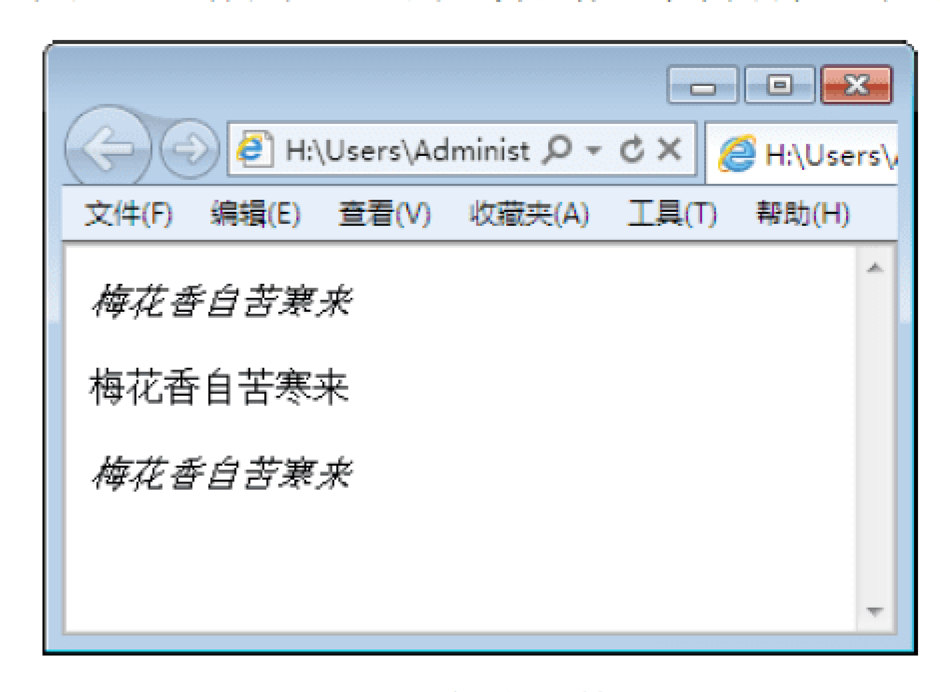


图 12-3　字体风格显示

12.1.4　加粗字体（font-weight）

通过设置字体的粗细，可以让文字显示不同的外观。通过 CSS 中的 font-weight 属性可以定义字体的粗细程度。其语法格式如下所示。

```
{font-weight:100~900|bold|bolder|lighter|normal;}
```

font-weight 属性有 13 个有效值，分别是 bold、bolder、lighter、normal、100~900。如果没有设置该属性，则使用其默认值 normal。属性值设置为 100~900，值越大，加粗的程度就越高。其具体含义如表 12-3 所示。

表 12-3　font-weight 属性表

属性值	描述
bold	定义粗体字体
bolder	定义更粗的字体，相对值
lighter	定义更细的字体，相对值
normal	默认，标准字体

浏览器默认的字体粗细是 400，另外也可以通过参数 lighter 和 bolder 使得字体在原有基础上显得更细或更粗。

【例 12.4】（实例文件：ch12\12.4.html）

```
<html>
<body>
    <p style="font-weight:bold">那年落叶风零的秋季(bold)</p>
    <p style="font-weight:bolder">彼此在茫茫人海相遇(bolder)</p>
    <p style="font-weight:lighter">不是天时，地利(lighter)</p>
    <p style="font-weight:normal">而是缘分前世的累积(normal)</p>
    <p style="font-weight:100">从那时起(100)</p>
    <p style="font-weight:400">一颗漂泊的心，不再寻寻觅觅(400)</p>
    <p style="font-weight:900">空旷的心房，不再冷冷清清(900)</p>
</body>
</html>
```

在 IE 9.0 中浏览效果如图 12-4 所示，可以看到文字居中并以不同方式加粗，其中使用了关键字加粗和数值加粗。

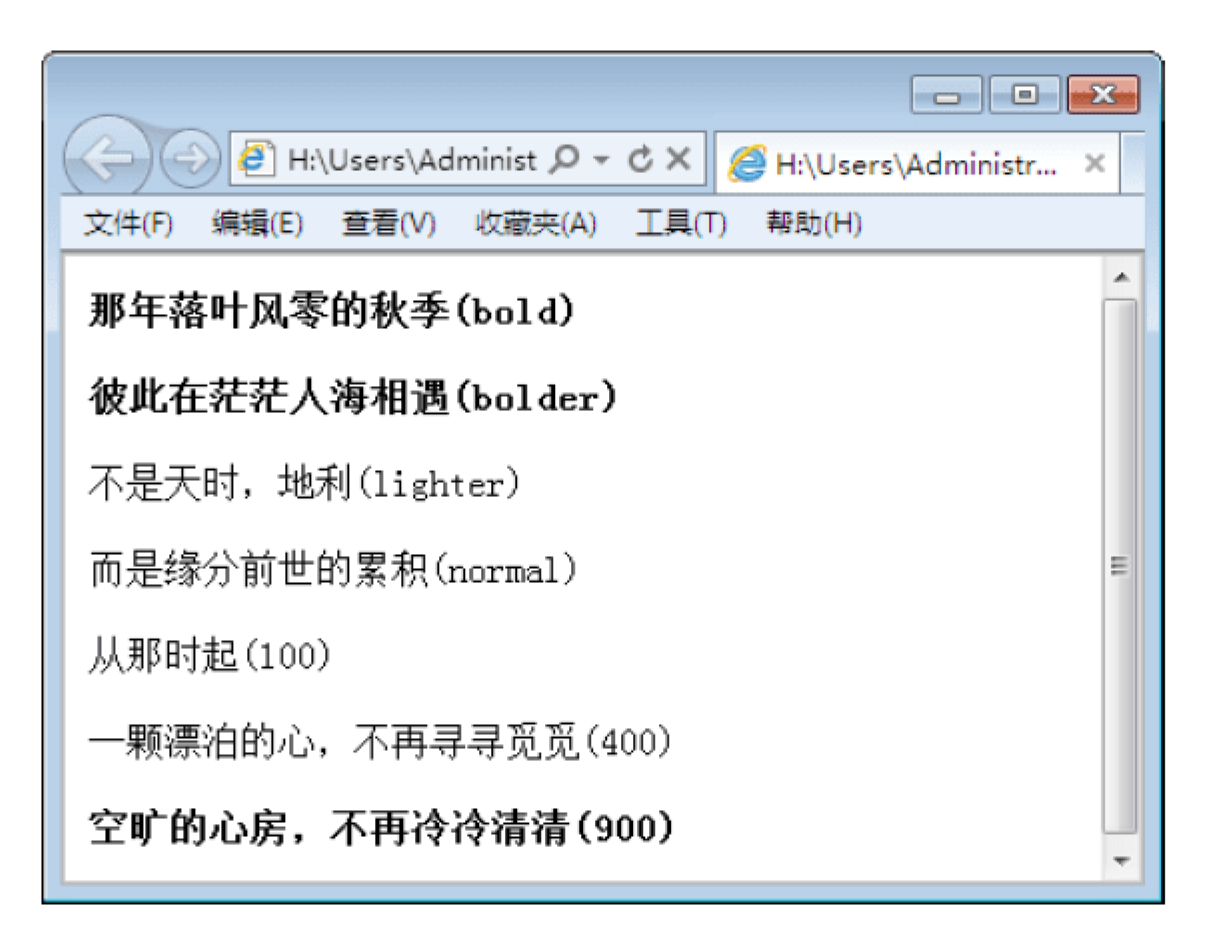


图 12-4 字体粗细显示

12.1.5 小写字母转为大写字母（font-variant）

font-variant 属性设置大写字母的字体显示文本，这意味着所有的小写字母均会被转换为大写，但是所有使用大写字体的字母与其余文本相比，其字体尺寸更小。在 CSS 中，其语法格式如下所示。

```
font-variant : normal | small-caps |inherit
```

font-variant 有 3 个属性值，分别是 normal、inherit 和 small-caps。其具体含义如表 12-4 所示。

表 12-4 font-variant 属性表

属性值	描述
normal	默认值。浏览器会显示一个标准的字体
small-caps	浏览器会显示小型大写字母的字体
inherit	规定应该从父元素继承 font-variant 属性的值

【例 12.5】（实例文件：ch12\12.5.html）

```
<html>
<body>
<p style="font-variant:normal">Happy BirthDay to You</p>
<p style="font-variant:small-caps">Happy BirthDay to You</p>
</body>
</html>
```

在 IE 9.0 中浏览效果如图 12-5 所示，可以看到字母以大写形式显示。

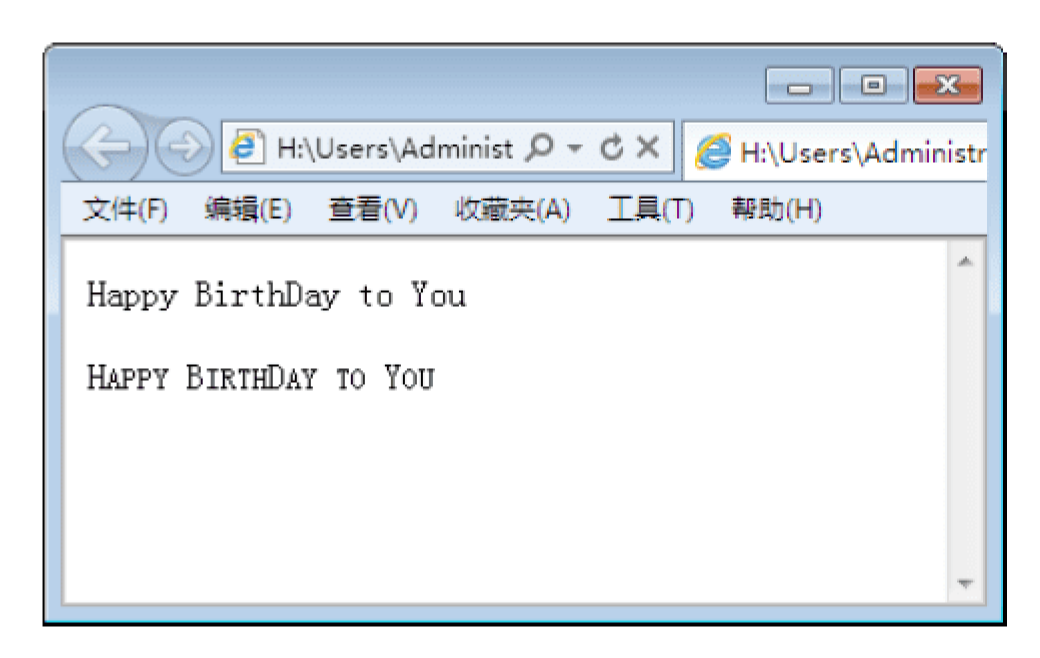


图 12-5　字母大小写转换

12.1.6　字体复合属性（font）

在设计网页时，为了使网页布局合理且文本规范，对字体的设计需要使用多种属性，例如定义字体粗细和字体大小。但是，分别书写多个属性相对比较麻烦，CSS 样式表提供的 font 属性就解决了这一问题。

font 属性可以一次性使用多个属性的属性值定义文本字体。其语法格式如下所示。

```
{font:font-style font-variant font-weight font-szie font-family}
```

font 属性中的属性排列顺序是 font-style、font-variant、font-weight、font-size 和 font-family，各个属性的属性值之间使用空格隔开，但是，如果 font-family 属性要定义多个属性值，则需使用逗号（,）隔开。

属性排列中，font-style、font-variant 和 font-weight 这 3 个属性值是可以自由调换的。而 font-size 和 font-family 则必须按照固定的顺序出现，而且还必须都出现在 font 属性中。如果这两者的顺序不对，或缺少一个，那么，整个样式规则可能就会被忽略。

【例 12.6】（实例文件：ch12\12.6.html）

```
<html>
<style type=text/css>
p{
    font:normal small-caps bolder 20pt "Cambria","Times New Roman",宋体
}
</style>
<body>
<p>
读书和学习是在别人思想和知识的帮助下，建立起自己的思想和知识。
</p>
</body>
</html>
```

在 IE 9.0 中浏览效果如图 12-6 所示，可以看到文字被设置成宋体并加粗。

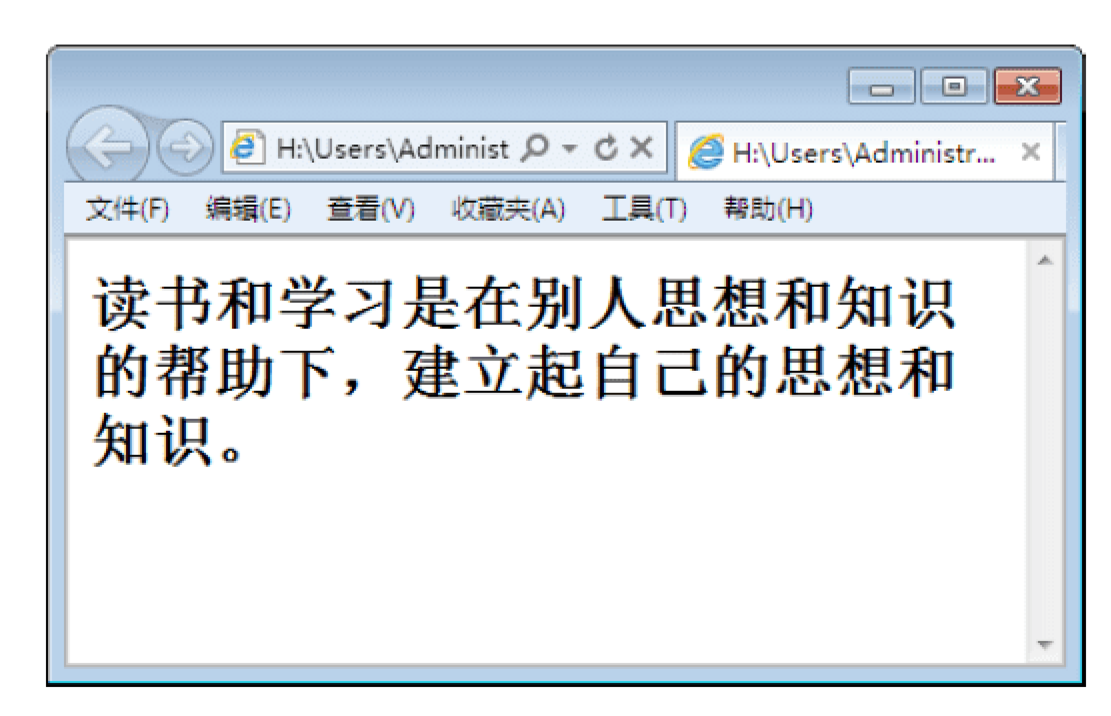


图 12-6　复合属性的显示

12.1.7　字体颜色（color）

没有色彩的网页是枯燥而没有生机的，这就意味着优秀的网页设计者不仅能够合理安排页面布局，而且还要具有一定的色彩搭配能力，这样才能够使网页更加精美、更具表现力，并给浏览者以亲切感。

在 CSS 样式中，通常使用 color 属性来定义颜色。其属性值通常使用下面的方式设定，如表 12-5 所示。

表 12-5　color 属性表

属性值	描述
color_name	规定颜色值为颜色名称的颜色（例如 red）
hex_number	规定颜色值为十六进制值的颜色（例如#ff0000）
rgb_number	规定颜色值为 rgb 代码的颜色（例如 rgb(255,0,0)）
inherit	规定应该从父元素继承颜色

【例 12.7】（实例文件：ch12\12.7.html）

```
<html>
<head>
<style type="text/css">
body {color:red}
h1 {color:#00ff00}
p.ex {color:rgb(0,0,255)}
</style>
</head>
<body>
<h1>这是标题 1</h1>
<p>这是一段普通的段落。请注意，该段落的文本是红色的。在 body 选择器中定义了本页面中的默认文本颜色。
</p>
<p class="ex">该段落定义了 class="ex"。该段落中的文本是蓝色的。</p>
```

```
</body>
</html>
```

在 IE 9.0 中浏览效果如图 12-7 所示，可以看到文字以不同颜色显示，代码中采用了不同的颜色取值方式。

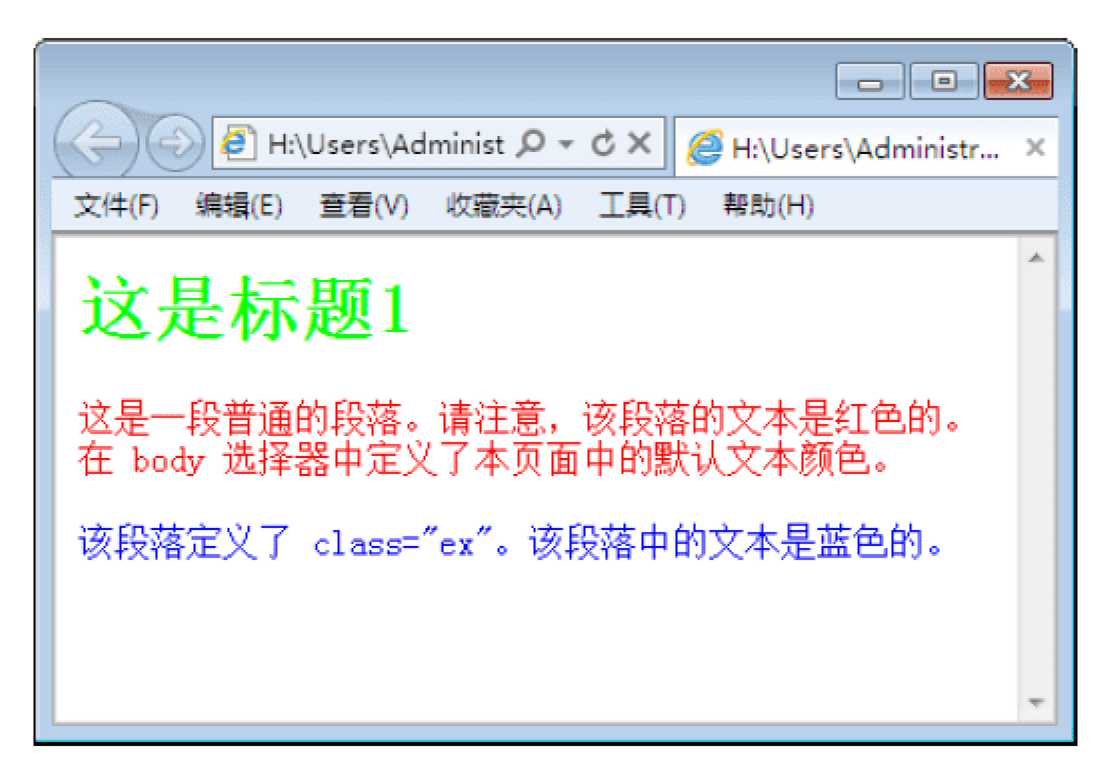


图 12-7　color 属性显示

12.2　CSS 段落样式

网页由文字组成，而用来表达同一个意思的多个文字组合，可以称为段落。段落是文章的基本单位，同样也是网页的基本单位。段落的放置与效果的显示会直接影响到页面的布局及风格。CSS 样式表提供了文本属性来实现对页面中段落文本的控制。

12.2.1　单词间隔（word-spacing）

如果设置合理，单词之间的间隔会为整个网页布局节省空间，还可以给人赏心悦目的感觉。在 CSS 中，可以使用 word-spacing 属性直接指定区域或者段落中字符之间的间隔。

word-spacing 属性用于设定词与词之间的间距，即增加或者减少词与词之间的间隔。其语法格式如下所示。

```
word-spacing : normal | length
```

其中属性值 normal 和 length 含义如表 12-6 所示。

表 12-6　单词间隔属性表

属性值	说明
normal	默认，定义单词之间的标准间隔
length	定义单词之间的固定宽带，可以接受正值或负值

【例 12.8】（实例文件：ch12\12.8.html）

```
<html>
<body>
```

```
<p style="word-spacing:normal">Welcome to my home</p>
<p style="word-spacing:15px">Welcome to my home</p>
<p style="word-spacing:15px">欢迎来中国旅游</p>
</body>
</html>
```

在 IE 9.0 中浏览效果如图 12-8 所示，可以看到段落中的单词以不同间隔显示。

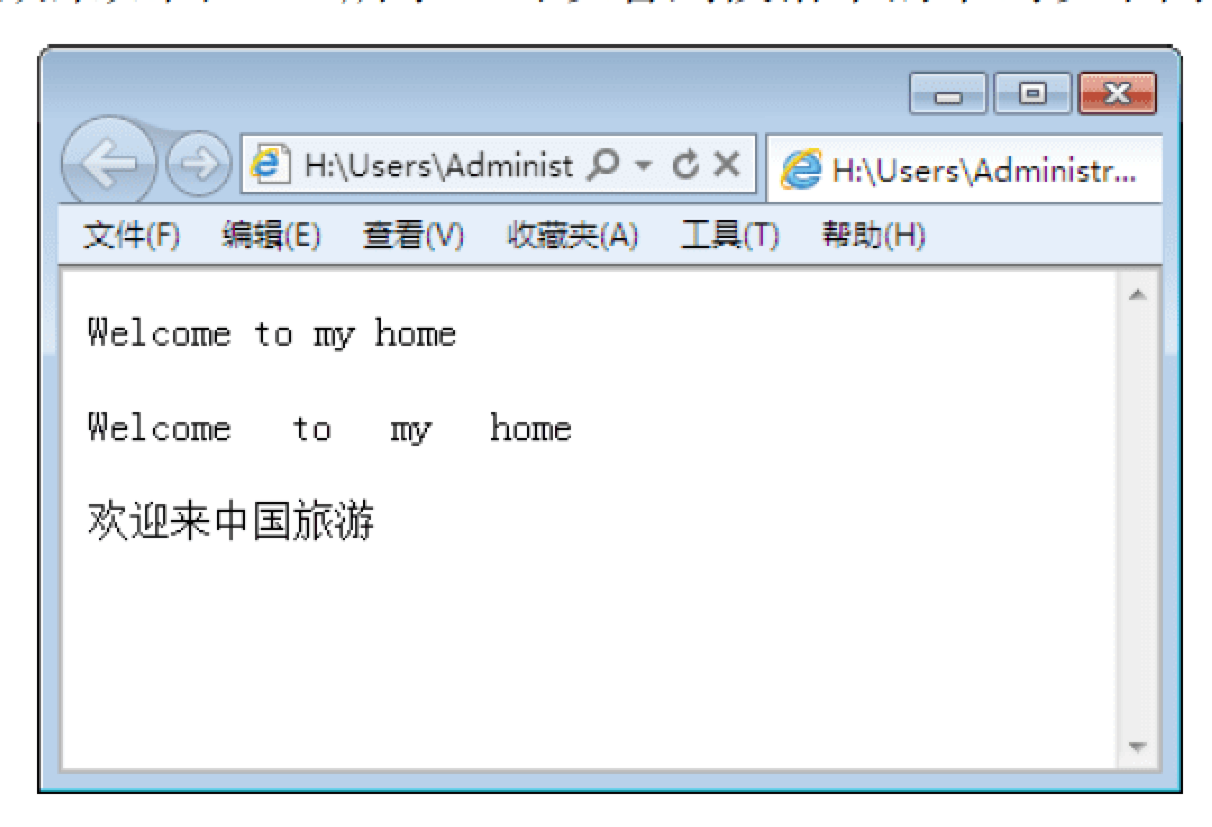


图 12-8　设定词间隔显示



从上面显示的结果可以看出，word-spacing 属性不能用于设定文字之间的间隔。

12.2.2　字符间隔（letter-spacing）

在一个网页中，还可能涉及多个字符文本，将字符文本之间的间距设置得和词间隔保持一致，进而保持页面的整体行，是网页设计者必须完成的。词与词之间可以通过 word-spacing 属性进行设置，那么字符之间使用什么设置呢？

在 CSS 中，可以通过 letter-spacing 属性来设置字符文本之间的距离。即在文本字符之间插入多少空间，这里允许使用负值，这会让字母之间更加紧凑。其语法格式如下所示。

```
letter-spacing : normal | length
```

其属性值含义如表 12-7 所示。

表 12-7　字符间隔属性表

属性值	说明
normal	默认间隔，即以字符之间的标准间隔显示
length	由浮点数字和单位标识符组成的长度值，允许为负值

【例 12.9】（实例文件：ch12\12.9.html）

```
<html>
<body>
```

```
<p style=" letter-spacing:normal">Welcome to my home</p>
<p style=" letter-spacing:5px">Welcome to my home</p>
<p style="letter-spacing:1ex">这里的字间距是 1ex</p>
<p style="letter-spacing:-1ex">这里的字间距是-1ex</p>
<p style="letter-spacing:1em">这里的字间距是 1em</p>
</body>
</html>
```

在 IE 9.0 中浏览效果如图 12-9 所示，可以看到文字间距以不同大小显示。

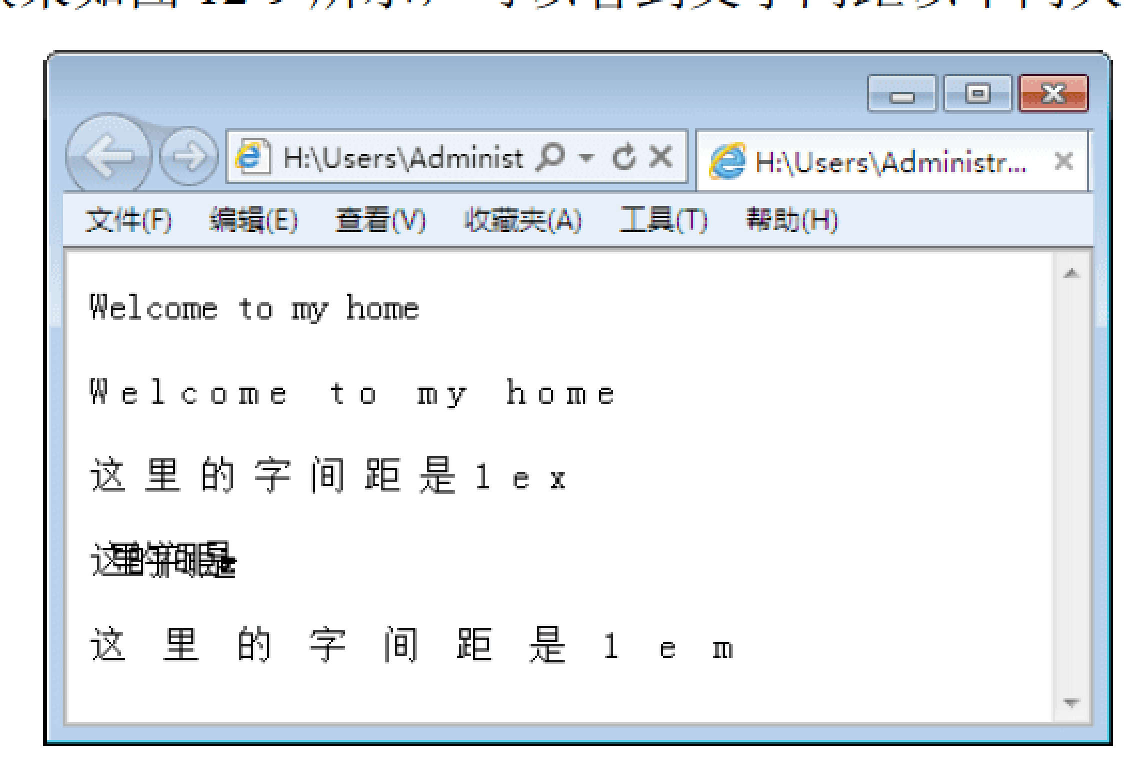


图 12-9　字间距效果



当设置的字间距为-1ex 时，文字就会粘到一块。

12.2.3　文字修饰（text-decoration）

在编辑网页文本时，有的文字需要突出重点，这时可以增加下划线，或者增加顶划线和删除线效果，从而吸引读者的眼球。在 CSS 中，text-decoration 属性是文本修饰属性，该属性可以为页面提供多种文本的修饰效果，如下划线、删除线、闪烁等。

text-decoration 属性语法格式如下所示。

```
text-decoration:none||underline||blink||overline||line-through
```

其属性值含义，如表 12-8 所示。

表 12-8　text-decoration 属性表

属性值	描述
none	默认值，对文本不进行任何修饰
underline	下划线
overline	上划线
line-through	删除线
blink	闪烁

【例 12.10】（实例文件：ch12\12.10.html）

```
<html>
<body>
  <p style="text-decoration:none">《清明》！</p>
  <p style="text-decoration:underline">清明时节雨纷纷，</p>
  <p style="text-decoration:overline">路上行人欲断魂。</p>
  <p style="text-decoration:line-through">借问酒家何处有，</p>
  <p style="text-decoration:blink">牧童遥指杏花村。</p>
</body>
</html>
```

在 IE 9.0 中显示效果如图 12-10 所示。可以看到段落中出现了下划线、上划线和删除线等。

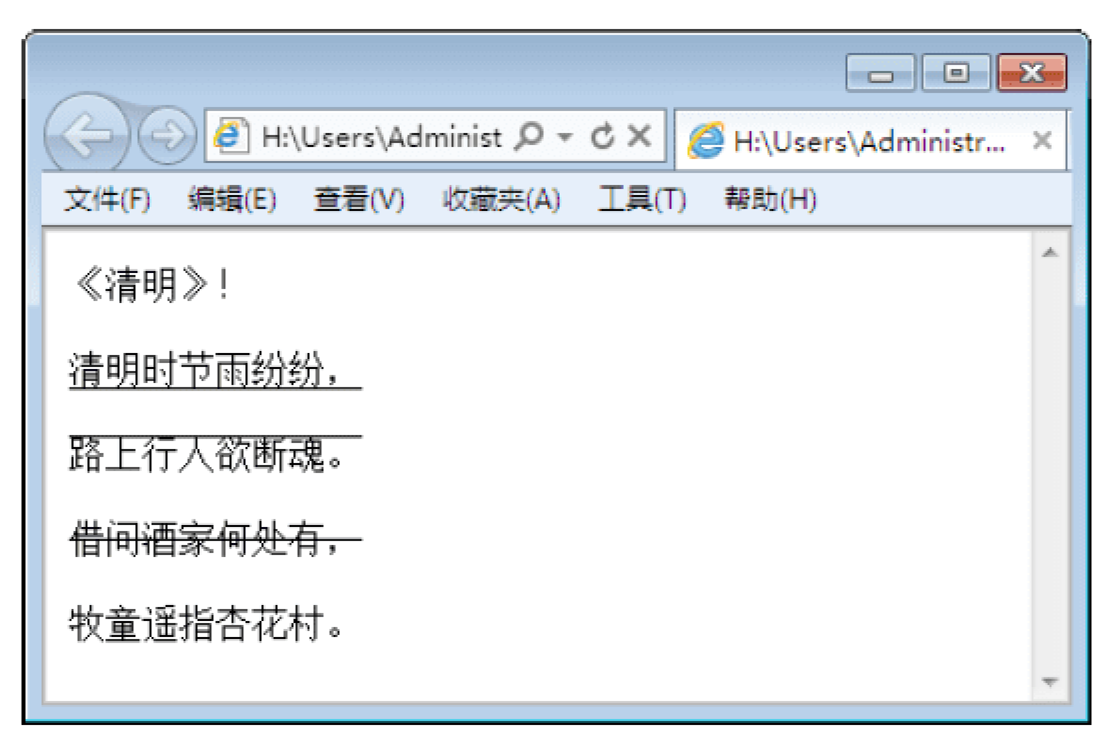


图 12-10 文本修饰显示



blink 闪烁效果只有 Mozilla 和 Netscape 浏览器支持，而 IE 和其他浏览器（如 Opera）都不支持该效果。

12.2.4 垂直对齐方式（vertial-align）

在网页文本编辑中，对齐有很多方式，字行排在一行的中央位置叫“居中”，文章的标题和表格中的数据一般都居中排。有时还要求文字垂直对齐，即文字顶部对齐，或者底部对齐。

在 CSS 中，可以直接使用 vertical-align 属性来定义，该属性用来设定垂直对齐方式。该属性定义行内元素的基线相对于该元素所在行的基线的垂直对齐。允许指定负长度值和百分比值。这会使元素降低而不是升高。在表单元格中，这个属性会设置单元格框中的单元格内容的对齐方式。

vertical-align 属性语法格式如下所示。

```
{vertical-align:属性值}
```

vertical-align 属性值有 8 个预设值可使用，也可以使用百分比。这 8 个预设值如表 12-9 所示。

表 12-9 vertical-align 属性表

属性值	说明
baseline	默认。元素放置在父元素的基线上
sub	垂直对齐文本的下标
super	垂直对齐文本的上标
top	把元素的顶端与行中最高元素的顶端对齐
text-top	把元素的顶端与父元素字体的顶端对齐
middle	把此元素放置在父元素的中部
bottom	把元素的顶端与行中最低的元素的顶端对齐
text-bottom	把元素的底端与父元素字体的底端对齐
%	使用 "line-height" 属性的百分比值来排列此元素。允许使用负值

【例 12.11】（实例文件：ch12\12.11.html）

```
<html>
<body>
<p>
    世界杯<b style=" font-size:8pt;vertical-align:super">2014</b>！
    中国队<b style="font-size: 8pt;vertical-align: sub">[注]</b>！
    加油！<img src="1.gif" style="vertical-align: baseline">
</p>
<p><img src="2.gif" style="vertical-align:middle"/>
    世界杯！中国队！加油！<img src="1.gif" style="vertical-align:top">
</p>
<hr/>
<p ><img src="2.gif" style="vertical-align:middle"/>
    世界杯！中国队！加油！<img src="1.gif" style="vertical-align:text-top">
</p>
<p><img src="2.gif" style="vertical-align:middle"/>
    世界杯！中国队！加油！<img src="1.gif" style="vertical-align:bottom">
</p>
<hr/>
<p ><img src="2.gif" style="vertical-align:middle"/>
    世界杯！中国队！加油！<img src="1.gif" style="vertical-align:text-bottom">
</p>
<p>
    世界杯<b style=" font-size:8pt;vertical-align:100%">2008</b>！
    中国队<b style="font-size: 8pt;vertical-align: -100%">[注]</b>！
    加油！<img src="1.gif" style="vertical-align: baseline">
</p>
</body>
```

```
</html>
```

在 IE 9.0 中浏览效果如图 12-11 所示，可以看到文字在垂直方向以不同的对齐方式显示。

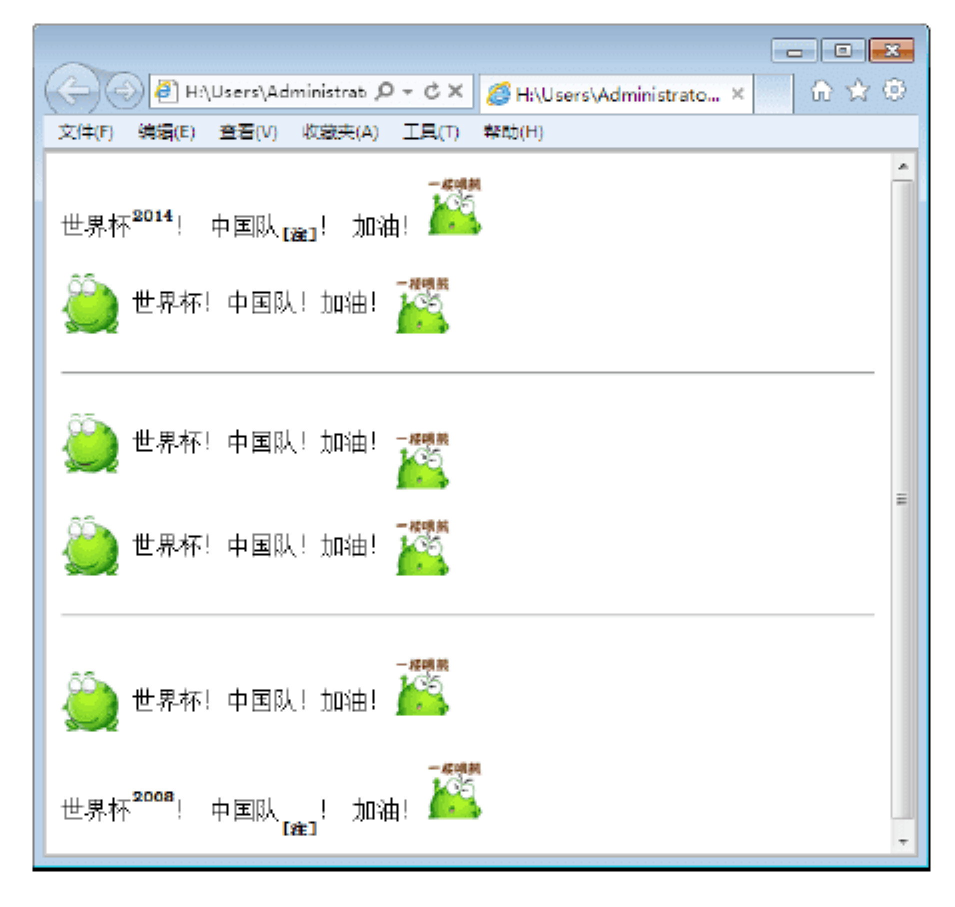


图 12-11　垂直对齐显示

从上面的实例中可以看出，上下标在页面中的数学运算或注释标号使用得比较多。顶端对齐有两种参照方式，一种是参照整个文本块，一种是参照文本。底部对齐同顶端对齐方式相同，分别参照文本块和文本块中包含的文本。



vertical-align 属性值还能使用百分比来设定垂直高度，该高度具有相对性的，它是基于行高的值来计算的。而且百分比还能使用正负号，正百分比使文本上升，负百分比使文本下降。

12.2.5　文本转换（text-transform）

根据需要，将小写字母转换为大写字母，或者将大写字母转换小写，在文本编辑中都是很常见的。在 CSS 样式中，text-transform 属性可用于设定文本字体的大小写转换。text-transform 属性语法格式如下所示。

```
text-transform : none | capitalize | uppercase | lowercase
```

其属性值含义，如表 12-10 所示。

表 12-10　text-transform 属性表

属性值	说明
none	无转换发生
capitalize	将每个单词的第一个字母转换成大写，其余无转换发生
uppercase	转换成大写
lowercase	转换成小写

因为文本转换属性仅作用于字母型文本，相对来说比较简单。

【例 12.12】（实例文件：ch12\12.12.html）

```
<html>
<body style="font-size:15pt; font-weight:bold">
  <p style="text-transform:none">welcome to beijing</p>
  <p style="text-transform:capitalize">welcome to beijing</p>
  <p style="text-transform:lowercase">WELCOME TO BEIJING</p>
  <p style="text-transform:uppercase">welcome to beijing</p>
</body>
</html>
```

在 IE 9.0 中浏览效果如图 12-12 所示。

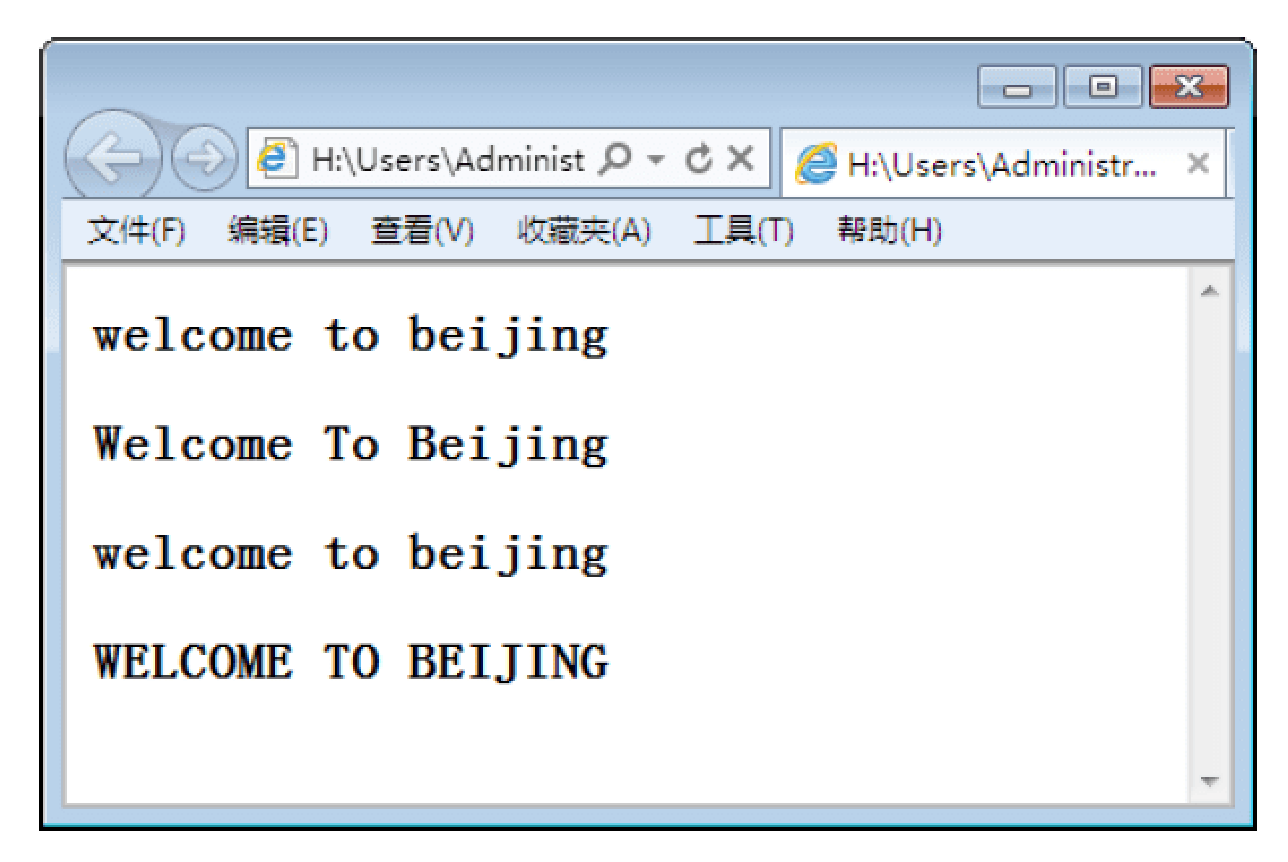


图 12-12　大小写字母转换显示

12.2.6　水平对齐方式（text-align）

一般情况下，居中对齐适用于标题类文本，其他对齐方式可以根据页面布局来选择使用。根据需要，可以设置多种对齐，例如水平方向上的居中、左对齐、右对齐或者两端对齐等。在 CSS 中，可以通过 text-align 属性进行设置。

text-align 属性用于定义对象文本的对齐方式，text-align 语法格式如下所示。

```
{ text-align: sTextAlign }
```

其属性值含义如表 12-11 所示。

表 12-11　text-align 属性表

属性值	说明
left	文本向行的左边缘对齐。在垂直方向的文本中，文本在 left-to-right 模式下向开始边缘对齐
right	文本向行的右边缘对齐。在垂直方向的文本中，文本在 left-to-right 模式下向结束边缘对齐

（续表）

属性值	说明
Center	文本在行内居中对齐
justify	文本根据 text-justify 的属性设置方法分散对齐。即两端对齐，均匀分布
match-parent	继承父元素的对齐方式，但有个例外：继承的 start 或者 end 值是根据父元素的 direction 值进行计算的，因此计算的结果可能是 left 或者 right
inherit	继承父元素的对齐方式

在新增加的属性值中，start 和 end 属性值主要是针对行内元素的，即在包含元素的头部或尾部显示；而<string>属性值主要用于表格单元格中，将根据某个指定的字符对齐。

【例 12.13】（实例文件：ch12\12.13.html）

```
<html>
<body>
<h1 style="text-align:center">登幽州台歌</h1>
<h3 style="text-align:left">选自：</h3>
<h3 style="text-align:right">
   <img src="1.gif" />
   唐诗三百首</h3>
<p style="text-align:justify">
   前不见古人
   后不见来者
</p>
</body>
</html>
```

在 IE 9.0 中浏览效果如图 12-13 所示，可以看到文字在水平方向上以不同的对齐方式显示。

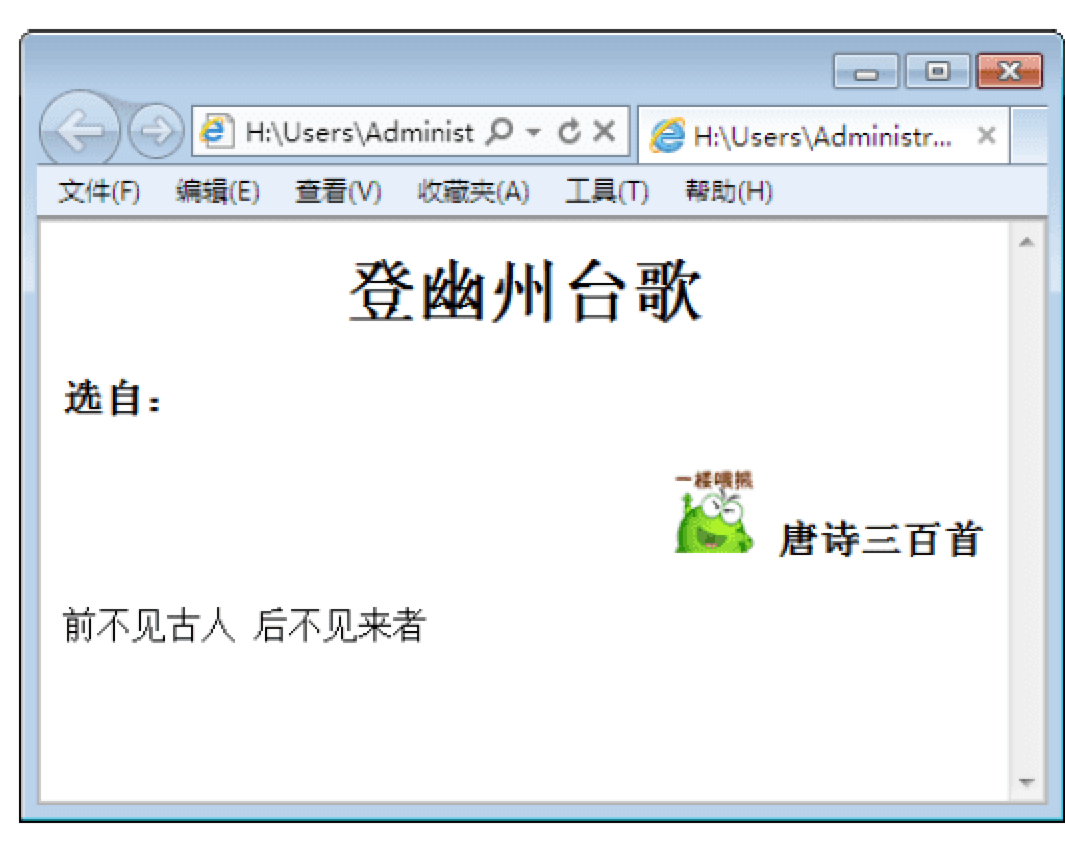


图 12-13　对齐效果图

text-align 属性只能用于文本块，而不能直接应用到图像标记<img>。如果要使图像同文本一样应用对齐方式，那么就必须将图像包含在文本块中。如例 12-13，由于向右对齐方式作用

于<h3>标记定义的文本块，图像包含在文本块中，所以图像能够同文本一样向右对齐。

12.2.7 文本缩进（text-indent）

在普通段落中，通常首行缩进两个字符，用来表示这是一个段落的开始。同样在网页的文本编辑中可以通过指定属性，来控制文本缩进。CSS 的 text-indent 属性就用来设定文本块中首行的缩进。

text-indent 属性语法格式如下所示。

```
text-indent : length
```

其中，length 属性值表示有百分比数字或有由浮点数字和单位标识符组成的长度值，允许为负值。可以这样认为，text-indent 属性可以定义两种缩进方式，一种是直接定义缩进的长度，另一种是定义缩进百分比。使用该属性，HTML 任何标记都可以使首行以给定的长度或百分比缩进。

【例 12.14】（实例文件：ch12\12.14.html）

```
<html>
<body>
<p style="text-indent:10mm">
    此处直接定义长度，直接缩进。
</p>
<p style="text-indent:10%">
    此处使用百分比，进行缩进。
</p>
</body>
</html>
```

在 IE 9.0 中浏览效果如图 12-14 所示，可以看到文字以首行缩进方式显示。

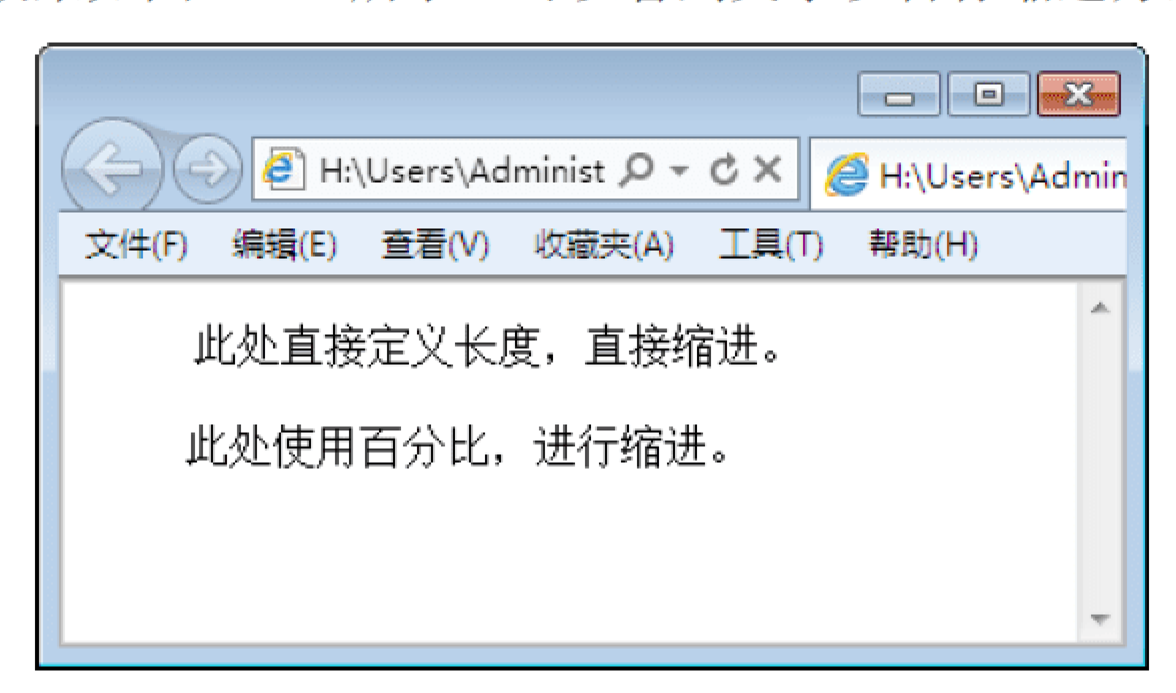


图 12-14 缩进显示窗口

如果上级标记定义了 text-indent 属性，那么子标记可以继承其上级标记的缩进长度。

12.2.8　文本行高（line-height）

在 CSS 中，line-height 属性用来设置行间距，即行高。其语法格式如下所示。

```
line-height : normal | length
```

其属性值的具体含义如表 12-12 所示。

表 12-12　line-height 属性表

属性值	说明
normal	默认行高，即网页文本的标准行高
length	百分比数字或由浮点数字和单位标识符组成的长度值，允许为负值。其百分比取值是基于字体的高度尺寸

【例 12.15】（实例文件：ch12\12.15.html）

```
<html>
<body>
  <div style="text-indent:10mm;">
    <p style="line-height:50px">
        世界杯（World Cup,FIFA World Cup），国际足联世界杯，世界足球锦标赛是世界上最高水平的足球比赛，与奥运会、F1 并称为全球三大顶级赛事。
    </p>    <p style="line-height:50%">
        世界杯（World Cup,FIFA World Cup），国际足联世界杯，世界足球锦标赛是世界上最高水平的足球比赛，与奥运会、F1 并称为全球三大顶级赛事。
    </p>
  </div>
</body>
</html>
```

在 IE 9.0 中浏览效果如图 12-15 所示，可以看到有段文字重叠在一起，即行高设置较小。

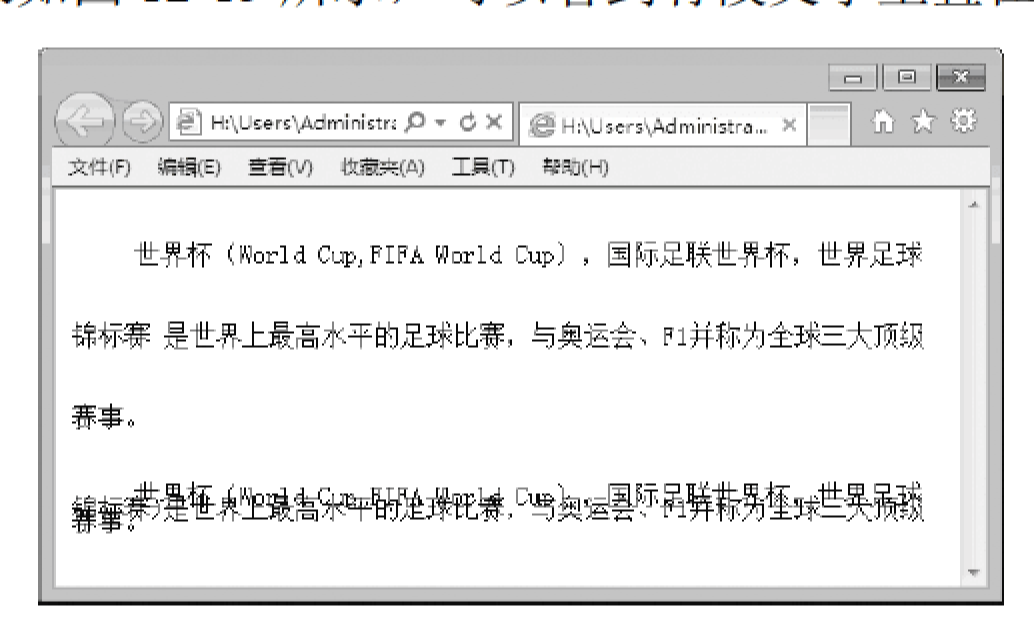


图 12-15　设定文本行高显示图

12.2.9　处理空白（white-sapce）

在文本编辑中，网页中有时需要包含一些不必要的制表符、换行符或者额外的空白符（多

于单词之间的一个标准的空格），这些符号统称为空白字符。通常情况下希望忽略这些额外的空白字符。浏览器可以自动完成此操作并按照一种适合窗口的方式布置文本。它会丢弃段落开头和结尾处任何额外的空白，并将单词之间的所有制表符、换行和额外的空白压缩（合并）成单一的空白字符。此外，当用户调整窗口大小时，浏览器会根据需要重新格式化文本以便匹配新的窗口尺寸。对于某些元素，可能会以某种方式特意格式化文本以便包含额外的空白字符，而不希望抛弃或压缩这些字符。

在 CSS 中，white-space 属性用于设置对象内空格字符的处理方式。white-space 属性对文本的显示有着重要的影响。在标记上应用 white-space 属性可以影响浏览器对字符串或文本间空白的处理方式。

white-space 属性语法格式如下所示。

```
white-space :normal | pre | nowrap | pre-wrap | pre-line
```

其属性值含义如表 12-13 所示。

表 12-13　空白属性表

属性值	说明
normal	默认。空白会被浏览器忽略
pre	空白会被浏览器保留。其行为方式类似 HTML 中的 <pre> 标签
nowrap	文本不会换行，文本会在在同一行上继续，直到遇到 标签为止
pre-wrap	保留空白符序列，但是正常地进行换行
pre-line	合并空白符序列，但是保留换行符
inherit	规定应该从父元素继承 white-space 属性的值

【例 12.16】（实例文件：ch12\12.16.html）

```
<html>
<body>
  <h1 style="color:red; text-align:center;white-space:pre">鸟鸣声里的春天！</h1>
  <div >
    <p style="white-space:nowrap;text-indent:10mm">
           一串悦耳的鸟叫，春天就来了。<br>
在乡村里，在一片一片的庄稼院里，最先体现春天到来的物事，便是那越来越嘹亮，越来越繁乱的鸟鸣。早晨里，天还未亮，空气还有些几分寒冷，一声长长的鸟鸣，便如一支铜号一般，呜呜噜噜撩开了春天的序幕。于是，在湿漉漉的泥土气息里，便有各种植物的种子，再也按捺不住的冲动，在鸟声里，嘣嘣嘣的爆出了嫩芽，甚至急不可耐地冲出了土面。
    </p>
    <p style="white-space:pre-wrap;text-indent:10mm">
          紧接着，黄鹂开始委婉缠绵的抒情，如二胡独奏曲中的《二泉映月》，如小提琴独奏曲中的《梁祝》，那么的一咏三叹，回环往复，直叫人荡气回肠，情不自禁而涌出满眼的热泪来。
                  在黄鹂、百灵鸟的吟唱声中，一棵又一棵的花木开始变绿，一朵一朵的樱花悄然绽放，而院墙内的那一枝红杏，早已探出了篱笆墙外。<br/>
```

```
                    黄鹂方才唱罢，在村庄的上空，在树林子里，在人家的土场上，一群花喜鹊便穿戴着黑白相间的朴素裙裾而闪亮登场，然后，便一天喜气的叽叽喳喳，叽叽喳喳叫起来。
            </p>
            <p style="white-space:pre-line;text-indent:10mm">
                    紧接着，黄鹂开始委婉缠绵的抒情，如二胡独奏曲中的《二泉映月》，如小提琴独奏曲中的《梁祝》，那么的一咏三叹，回环往复，直叫人荡气回肠，情不自禁而涌出满眼的热泪来。
                    在黄鹂、百灵鸟的吟唱声中，一棵又一棵的花木开始变绿，一朵一朵的樱花悄然绽放，而院墙内的那一枝红杏，早已探出了篱笆墙外。<br/>
                    黄鹂方才唱罢，在村庄的上空，在树林子里，在人家的土场上，一群花喜鹊便穿戴着黑白相间的朴素裙裾而闪亮登场，然后，便一天喜气的叽叽喳喳，叽叽喳喳叫起来。
            </p>
        </div>
    </body>
</html>
```

在 IE 9.0 中浏览效果如图 12-16 所示，可以看到文字处理空白的不同方式。

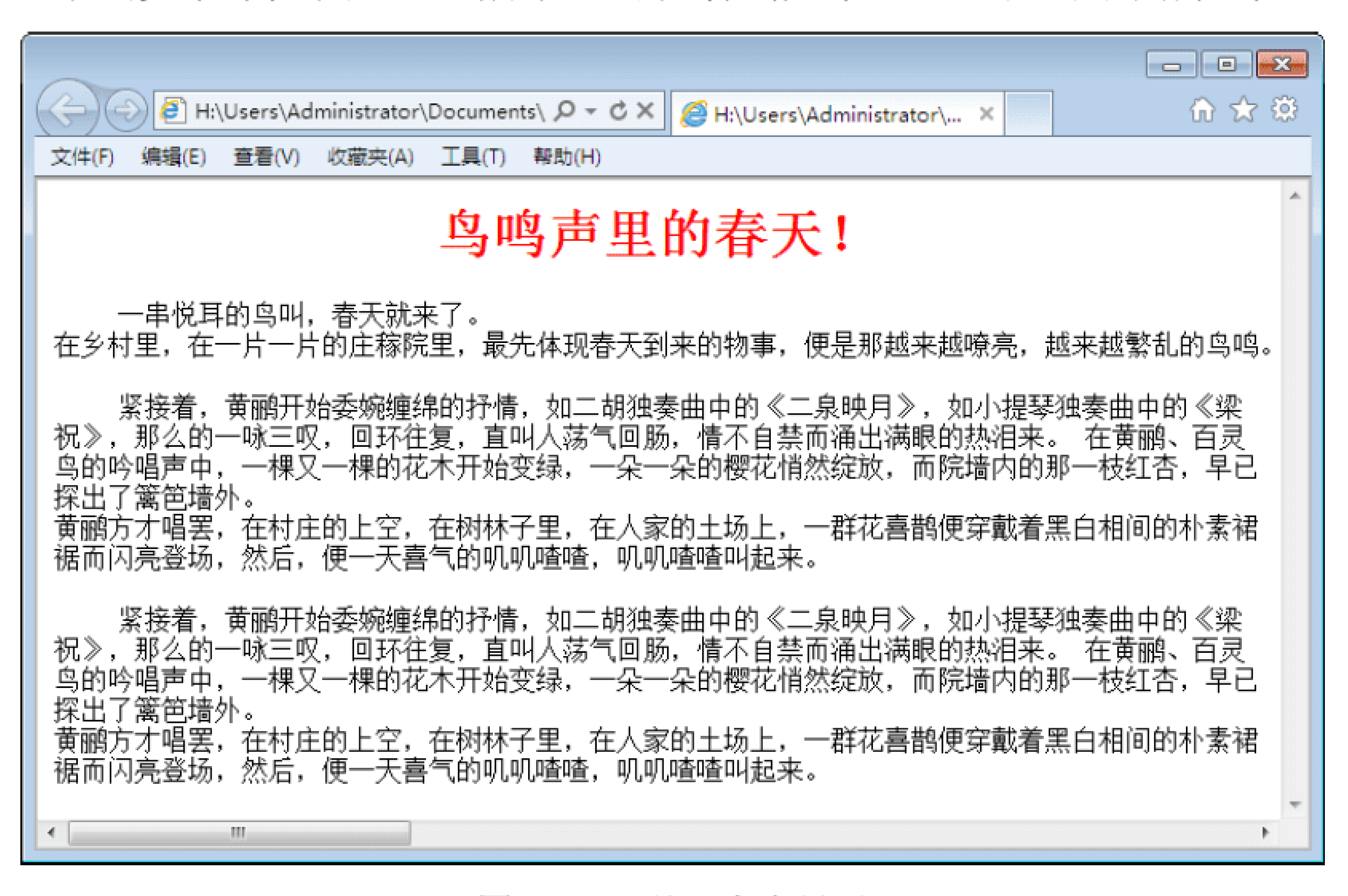


图 12-16　处理空白显示

12.2.10　文本反排（unicode-bidi 和 direction）

在网页文本编辑中，通常英语文档的基本方向是从左至右。如果文档中某一段的多个部分包含从右至左阅读的语言，则该语言的方向需要正确地显示为从右至左。此时可以通过 CSS 提供的两个属性 unicode-bidi 和 direction 解决文本反排的问题。

unicode-bidi 属性语法格式如下所示。

```
unicode-bidi : normal | bidi-override | embed
```

其属性值含义如表 12-14 所示。

表 12-14 unicode-bidi 属性表

属性值	说明
normal	默认值。元素不会打开一个额外的嵌入级别。对于内联元素，隐式的重新排序将跨元素边界起作用
bidi-override	与 embed 值相同，但除了这一点外：在元素内，重新排序依照 direction 属性严格按顺序进行。此值替代隐式双向算法
embed	元素将打开一个额外的嵌入级别。 direction 属性的值指定嵌入级别。重新排序在元素内是隐式进行的

direction 属性用于设定文本流的方向，其语法格式如下所示。

```
direction : ltr | rtl | inherit
```

其属性值含义如表 12-15 所示。

表 12-15 direction 属性表

属性值	说明
ltr	文本流从左到右
rtl	文本流从右到左
inherit	文本流的值不可继承

【例 12.17】（实例文件：ch12\12.17.html）

```
<html>
<head>
<style type="text/css">
a {color:#000;}
</style>
</head>
<body>
<h3>纯 CSS 反转字符串 使用了 direction 和 unicode-bidi</h3>
<div style=" direction:rtl; unicode-bidi:bidi-override; text-align:left">秋风吹不尽，总是玉关情。
</div>
</body>
</html>
```

在 IE 9.0 中浏览效果如图 12-17 所示，可以看到文字以反转形式显示。

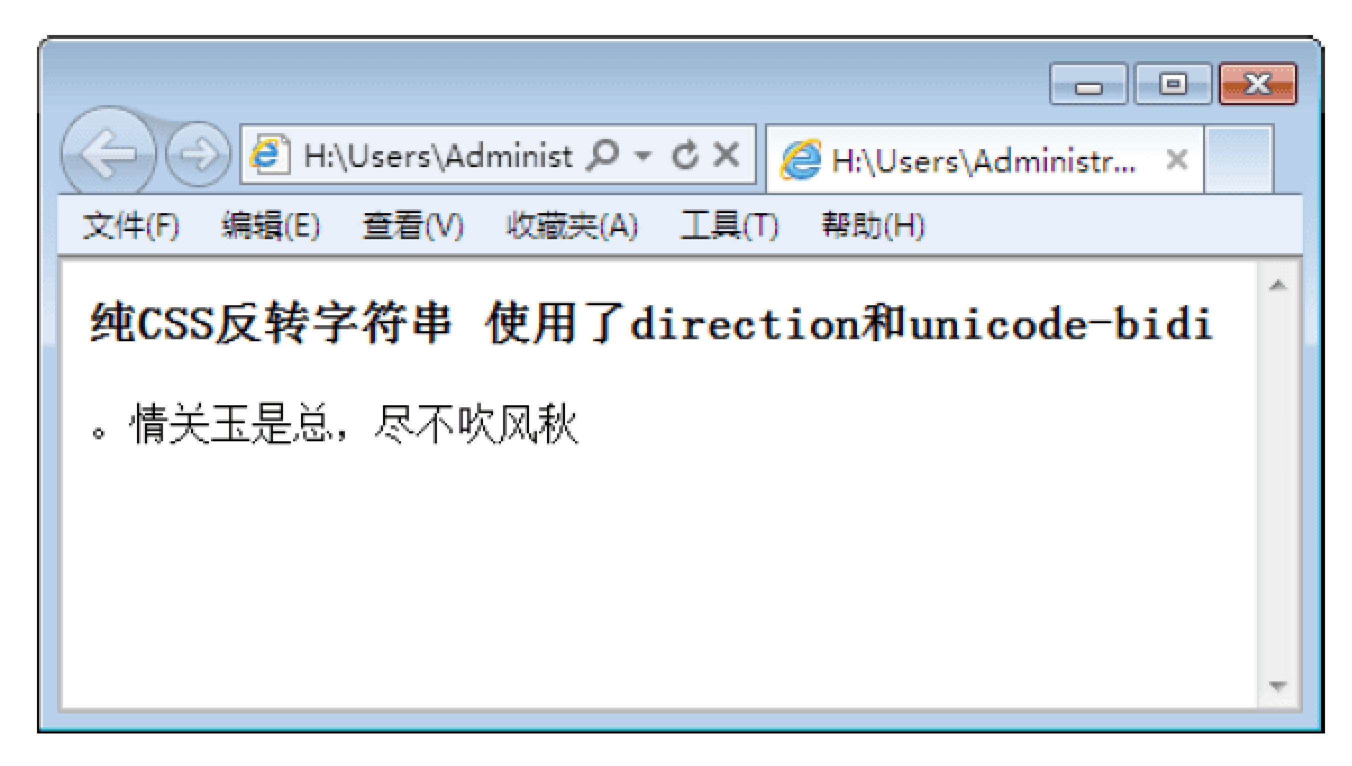


图 12-17 文本反转显示

12.4 综合实例——网页简单图文混排

在一个网页新闻中，出现最多就是文字和图片，二者放在一起，图文并茂，才能够生动地表达新闻主题。本实例将利用前面介绍的文本和段落属性，创建一个图片的简单混排。复杂的图片混排会在后面介绍。具体步骤如下所示。

01 分析需求。

本综合实例的要求如下：在网页的最上方显示出标题，标题下方是正文，在正文显示部分显示图片。在设计这个网页标题时，其方法同上面的实例相同。上述要求使用 CSS 样式属性实现。其实例效果如图 12-18 所示。



图 12-18 图文混排效果

02 分析布局并构建 HTML。

首先需要创建一个 HTML 页面，并用 DIV 将页面划分为两个层，一个是网页标题层，一个是正文部分。

03 导入 CSS 文件。

将 CSS 文件使用 link 方式导入到 HTML 页面中。此 CSS 页面定义了这个页面的所有样式，其导入代码如下所示：

```
<link href="12-19.css" rel="stylesheet" type="text/css" />
```

04 完成标题部分。

首先设置网页标题部分，创建一个 div，用来放置标题。其 HTML 代码如下所示：

```
<div>
<h1>四月秋语，演绎别类西海情缘</h1>
</div>
```

在 CSS 样式文件中，修饰 HTML 元素，其 CSS 代码如下所示：

```
h1{text-align:center;text-shadow:0.1em 2px 6px blue;font-size:18px;}
```

05 完成正文和图片部分。

下面设置网页正文部分，正文中包含了一个图片。其 HTML 代码如下所示：

```
<div>
<p>自你离开以后，从此就丢了温柔。等待在这雪山路漫长，听寒风呼啸依旧。一眼望不到边，风似刀割我的脸。等不到西海天际蔚蓝，无言着苍茫的高原。还记得你答应过我不会让我把你找不见，可你跟随那南归的候鸟飞得那么远，爱像风筝断了线，拉不住你许下的诺言。我在苦苦等待雪山之巅温暖的春天，等待高原冰雪融化之后归来的孤雁，爱再难以续情缘，回不到我们的从前-----四月秋语，演绎别类西海情缘。
</p><p>一首凄婉深情之歌谣，扰乱了晚春数日之心绪。沉重的笔墨延续歌词的悲哀，眼茫茫，心戚戚，泪成行。于每一个孤清的夜晚，在松软的枕上汪汇成深潭！</p>
<DIV class="im">
<img src="图 01.jpg">
</DIV>
<p>自你离开以后，从此就丢掉了笑颜。等待，在这春暖花开的季节，听候鸟南归，看春花烂漫。路漫长， 却依然望不到心中那抹湛蓝。一望无垠的天际，呈现灰蒙蒙一片，荒凉了心里所有的期许和祈盼！流年的秋季，离别的场景，还一一浮现于眼前。萧瑟的秋风吹散所有的缠缠绵绵，枫叶飘飘，落英缤纷，掩盖了一地的心碎！看着愈渐模糊的背影，欲想挽留却泛发无力，泪滑落，心凄然！任凭其颤颤巍巍的身影就这样渐行渐远....
</p>
</div>
```

CSS 样式代码如下所示：

```
p{text-indent:8mm;line-height:7mm;}
.im{width:300px; float:left; border:#000000 solid 1px;}
```

12.5 专家解惑

1. 某些字体为什么在别的电脑上不显示？

楷体很漂亮，草书也不逊色于宋体。但不是所有人的电脑都安装了这些字体。所以在设计网页时，不要为了追求漂亮美观，而采用一些比较新奇的字体。有时这样往往达不到效果。用最基本的字体，是最好的选择。

不要使用难于阅读的花哨字体。当然，某些字体可以让网页精彩纷呈。不过它们容易阅读吗？网页的主要目的是传递信息并易于读者阅读，我们应该让读者阅读过程舒服些。不要使用小号字体，虽然 Firefox 浏览器有放大功能，但如果必须放大才能看清一个网站的话，以后读者就再也不会去访问它了。

2. 网页中的空白处理吗？

注意不留空白。不要用图像、文本和不必要的动画来充斥网页，即使有足够的空间，在设计时也应该避免使用。

3. 文字和图片导航速度谁快呢？

通常使用文字做导航栏。文字导航不仅速度快，而且更稳定。例如，有些用户上网时会关闭图片。在处理文本时，除非特别需要，否则不要为普通文字添加下划线。因为用户需要识别哪些文字能够点击。

第 13 章

用CSS美化图片

网页中如果都是文字，会给浏览者枯燥的感觉，而一张恰如其分的图片，会为网页带来许多生趣。图片是直观、形象的，一张好的图片会给网页带来很高的点击率。在 CSS 中，定义了用来美化和设置图片的很多属性。

13.1　修饰图片

如果将多张图片直接放置到网页中，而不加修饰，会给人一种凌乱的感觉。通过 CSS 属性统一管理，可以定义多张图片的各种属性，例如图片边框、图片大小和缩放。

13.1.1　图片边框

在网页中放置一张图片时，可以使用<img>标记，在第 2 章中已经介绍过了。当图片显示之后，其边框是否显示，可以通过<img>标记中的描述标记 border 来设定。其示例代码为：

```
<img src="yueji.jpg" border="3">
```

通过 HTML 标记设置图片边框，其边框显示都是黑色的，并且风格比较单一，唯一能够设定的就是边框的粗细，而对边框样式基本上无能为力。这时可以采用 CSS 对边框进行美化。

在 CSS 中，可以使用 border-style 属性定义边框样式，即边框风格。例如可以将边框风格设置为点线式边框（dotted）、破折线式边框（dashed）、直线式边框（solid）、双线式边框（double）等。

【例 13.1】（实例文件：ch13\13.1.html）

```
<html>
<head>
<title>图片边框</title>
</head>
<body>
<img src="yueji.jpg" border="3" style="border-style:dotted">
<img src="yueji.jpg" border="3" style="border-style:dashed">
</body>
</html>
```

在 IE 9.0 中浏览效果如图 13-1 所示，可以看到网页中显示了两张图片，其边框分别为点线式和破折线式。

另外，如果需要单独定义边框一边的样式，可以使用 border-top-style 设定上边框样式、border-right-style 设定右边框样式、border-bottom-style 设定下边框样式和 border-left-style 设定左边框样式。

【例 13.2】（实例文件：ch13\13.2.html）

```
<html>
<head>
<title>图片边框</title>
</head>
<body>
<img src="yueji.jpg" border="3" style="border-top-style:dotted;border-right-style:insert;border-bottom-style:dashed;border-left-style:groove">
</body>
</html>
```

在 IE 9.0 中浏览效果如图 13-2 所示，可以看到网页显示了一张图片，图片的上边框、下边框、左边框和右边框分别以不同样式显示。

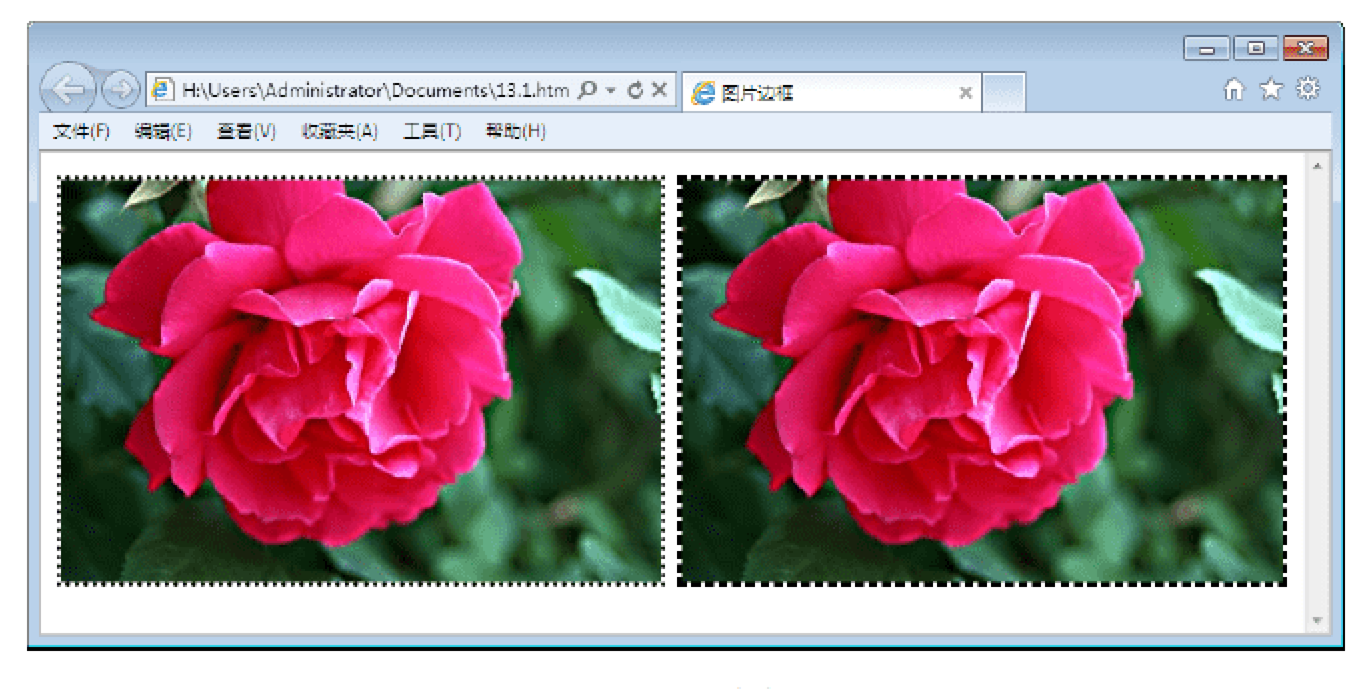


图 13-1　边框显示

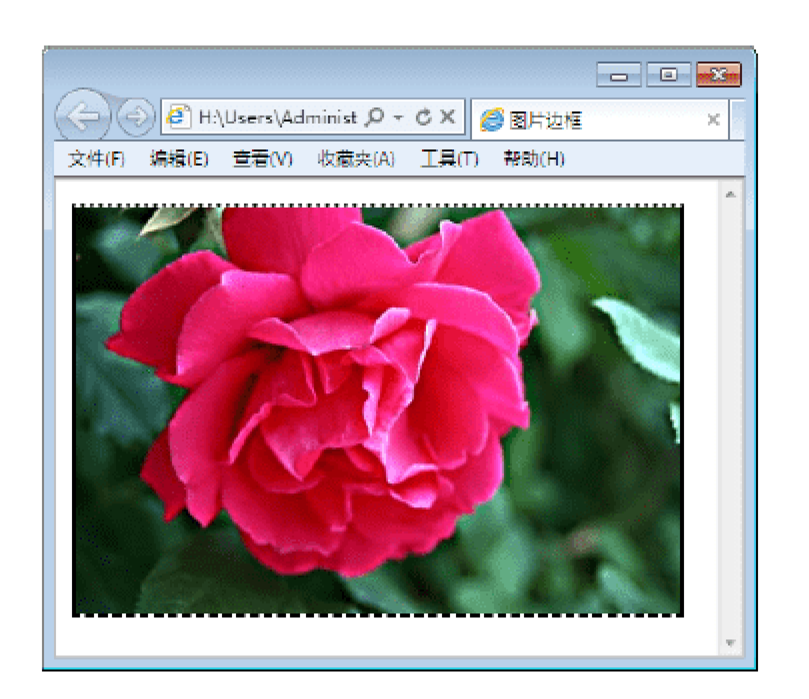


图 13-2　4 种样式边框显示

13.1.2　图片缩放

在网页上显示一张图片时，默认情况下都是以图片的原始大小显示。如果要对网页进行排版，通常情况下，还需要对图片大小进行重新设定。对于图片大小的设定，可以采用 3 种方式完成。

1. 使用描述标记 width 和 height

在 HTML 标记语言中，通过 img 的描述标记 height 和 width 可以设置图片大小。width 和 height 分别表示图片的宽度和高度，其中二者可以是数值或百分比，单位可以是像素。需要注意的是高度属性 heigth 和宽度属性 width 的设置要求相同。

【例 13.3】（实例文件：ch13\13.3.html）

```
<html>
<head>
<title>图片边框</title>
</head>
<body>
<img src="feng.jpg" width=200 height=120>
</body>
</html>
```

在 IE 9.0 中浏览效果如图 13-3 所示，可以看到网页中显示了一张图片，其宽度为 200 像素，高度为 120 像素。

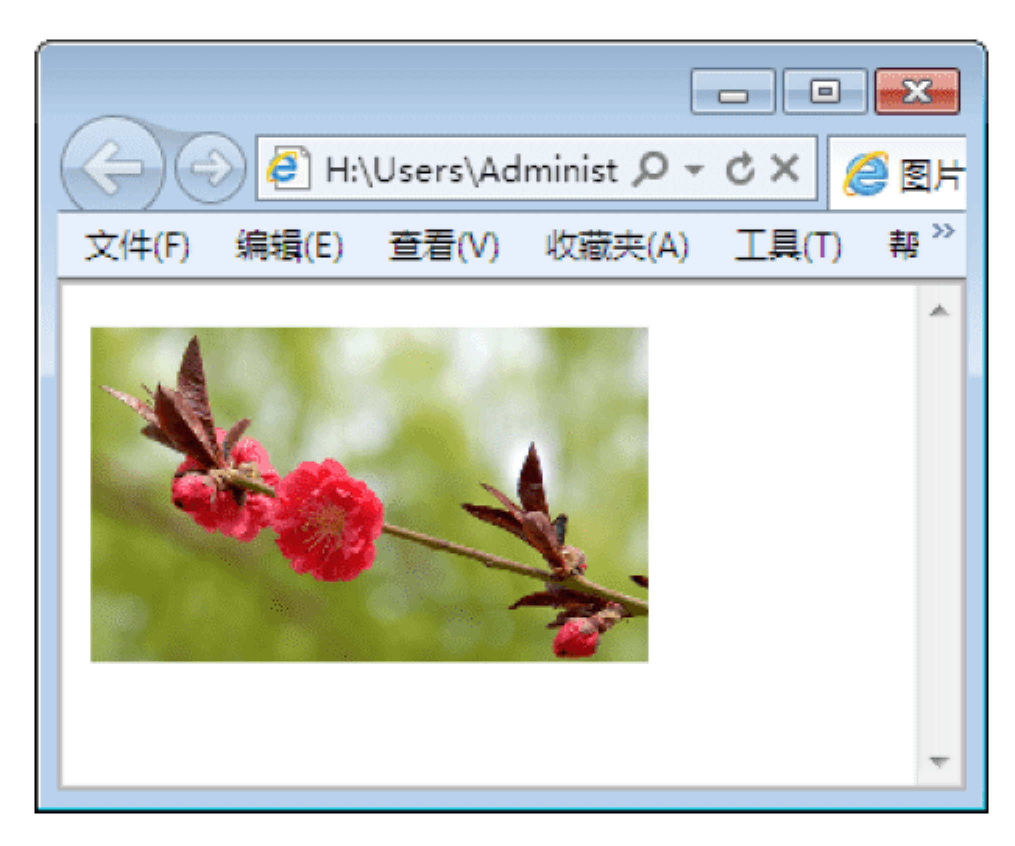


图 13-3　使用 HTML 标记设定图片大小

2．使用 max-width 和 max-height

max-width 和 max-height 分别用来设置图片宽度最大值和高度最大值。在定义图片大小时，如果图片默认尺寸超过了定义的大小，那么就以 max-width 所定义的宽度值显示，而图片高度将同比例变化，如果定义的是 max-height，依此类推。但是如果图片的尺寸小于最大宽度或者高度，那么图片就按原尺寸大小显示。max-width 和 max-height 的值一般是数值类型。

其语法格式如下所示：

```
img{
    max-height:180px;
}
```

【例 13.4】（实例文件：ch13\13.4.html）

```
<html>
<head>
<title>图片边框</title>
<style>
img{
    max-height:180px;
```

```
}
</style>
</head>
<body>
<img src="feng.jpg" >
</body>
</html>
```

在 IE 9.0 中浏览效果如图 13-4 所示，可以看到网页中显示了一张图片，其显示高度是 120 像素，宽度将做同比例缩放。

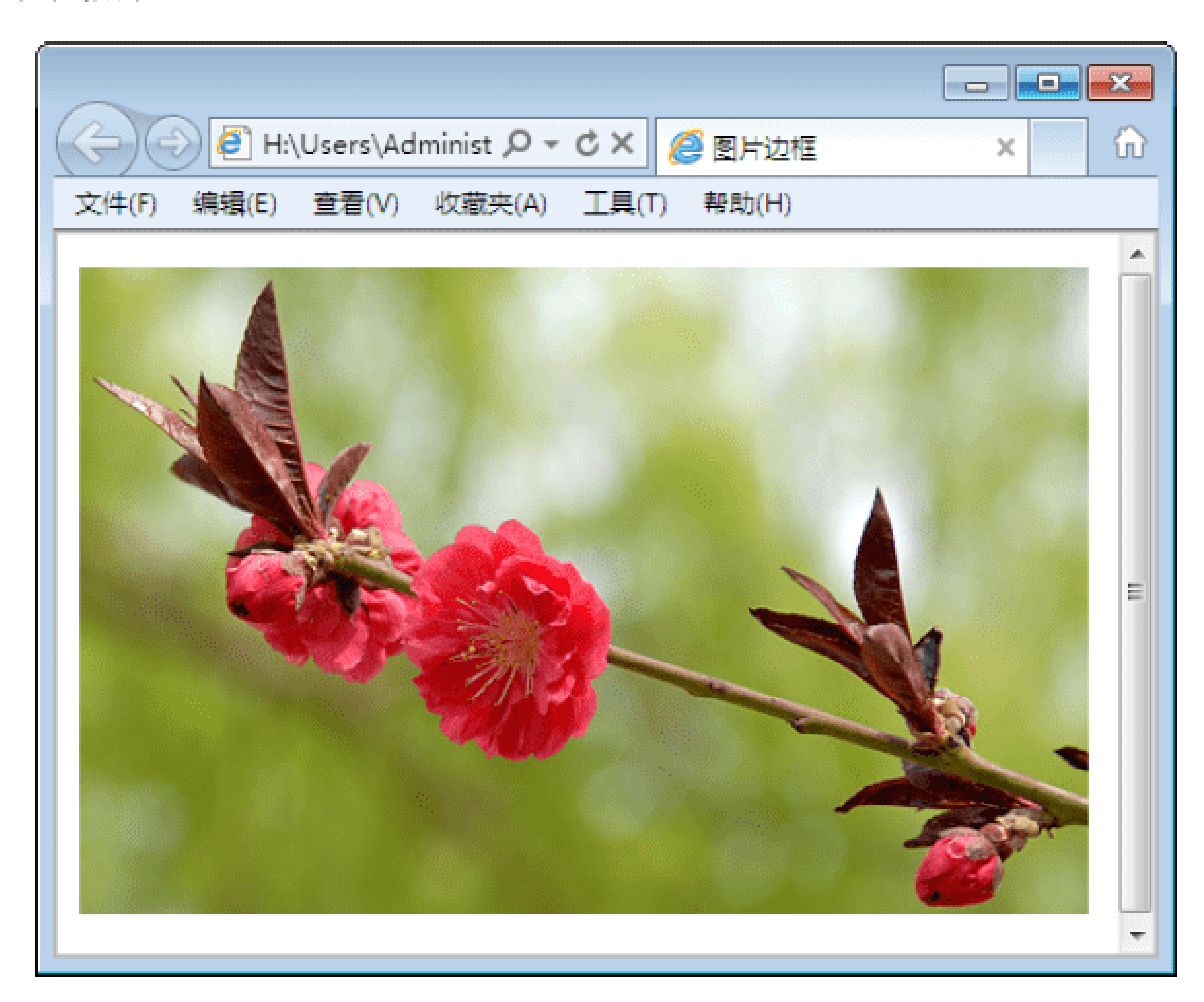


图 13-4　同比例缩放图片

在本例中，也可以只设置 max-width 来定义图片最大宽度，而让高度自动缩放。

3．使用 width 和 height 属性

在 CSS 中，可以使用属性 width 和 height 来设置图片宽度和高度，从而达到图片的缩放效果。

【例 13.5】（实例文件：ch13\13.5.html）

```
<html>
<head>
<title>图片边框</title>
</head>
<body>
<img src="feng.jpg" >
<img src="feng.jpg"   style="width:90px;height:100px" >
</body>
</html>
```

在 IE 9.0 中浏览效果如图 13-5 所示，可以看到网页中显示了两张图片，第一张图片以原大小显示，第二张图片以指定大小显示。

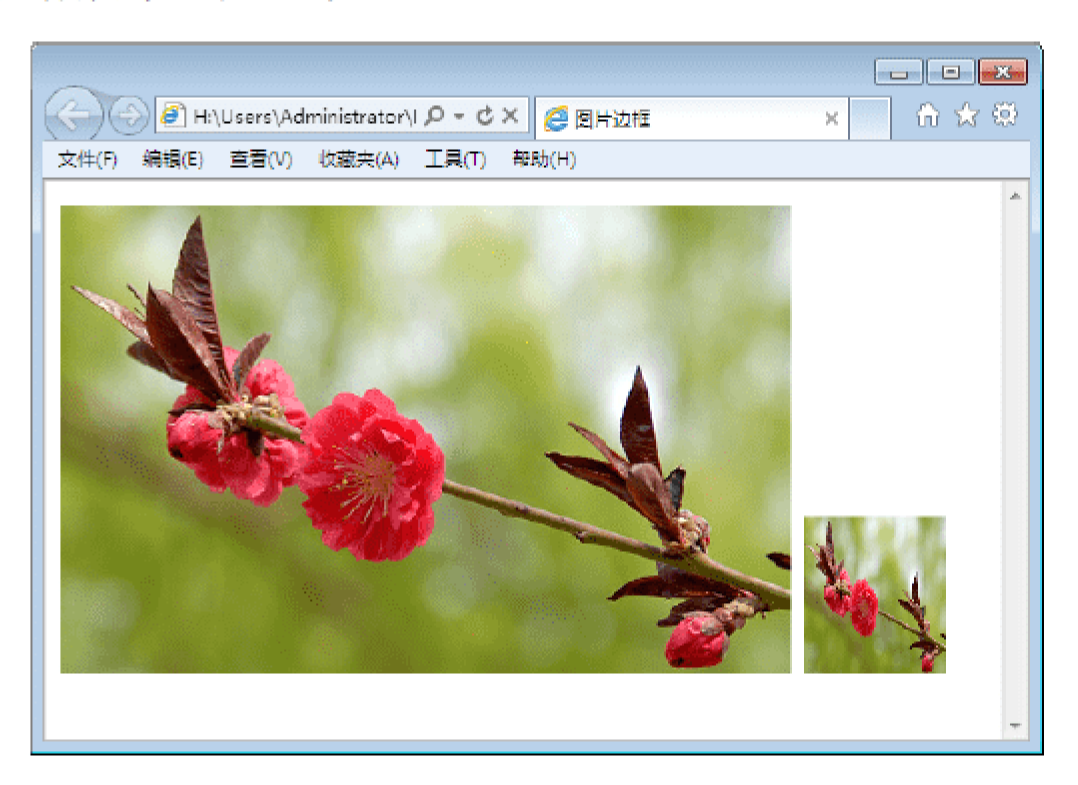


图 13-5　使用 CSS 指定图片大小



当仅仅设置了图片的 width 属性，而没有设置 height 属性时，图片本身会自动等纵横比例缩放，如果只设置了 height 属性也是同样的道理。只有当同时设置了 width 和 height 属性时才会不等比例缩放。

13.2　对齐图片

一个凌乱的图文网页是每一个浏览者都不喜欢看到的。而一个图文并茂、排版整洁的页面，更容易让网页浏览者接受。可见，图片的对齐方式是非常重要的。

13.2.1　横向对齐方式

所谓图片横向对齐，就是在水平方向上进行对齐，其对齐方式和文字对齐比较相似，也是有 3 种对齐方式，分别为左对齐、右对齐和居中对齐。

如果要定义图片的对齐方式，不能在样式表中直接定义图片样式，需要在图片的上一个标记级别，即父标记定义，让图片继承父标记的对齐方式。之所以这样定义父标记对齐方式，是因为图片本身没有对齐属性，需要使用 CSS 继承父标记的 text-align 来定义对齐方式。

【例 13.6】（实例文件：ch13\13.6.html）

```
<html>
<head>
<title>图片横向对齐</title>
</head>
<body>
<p style="text-align:left"><img src="mudan.jpg" style="max-width:140px;">图片左对齐</p>
<p style="text-align:center"><img src="mudan.jpg" style="max-width:140px;">图片居中对齐</p>
<p style="text-align:right"><img src="mudan.jpg" style="max-width:140px;">图片右对齐</p>
</body>
```

```
</html>
```

在 IE 9.0 中浏览效果如图 13-6 所示，可以看到网页上显示了 3 张图片，大小一样，但对齐方式分别是左对齐、居中对齐和右对齐。

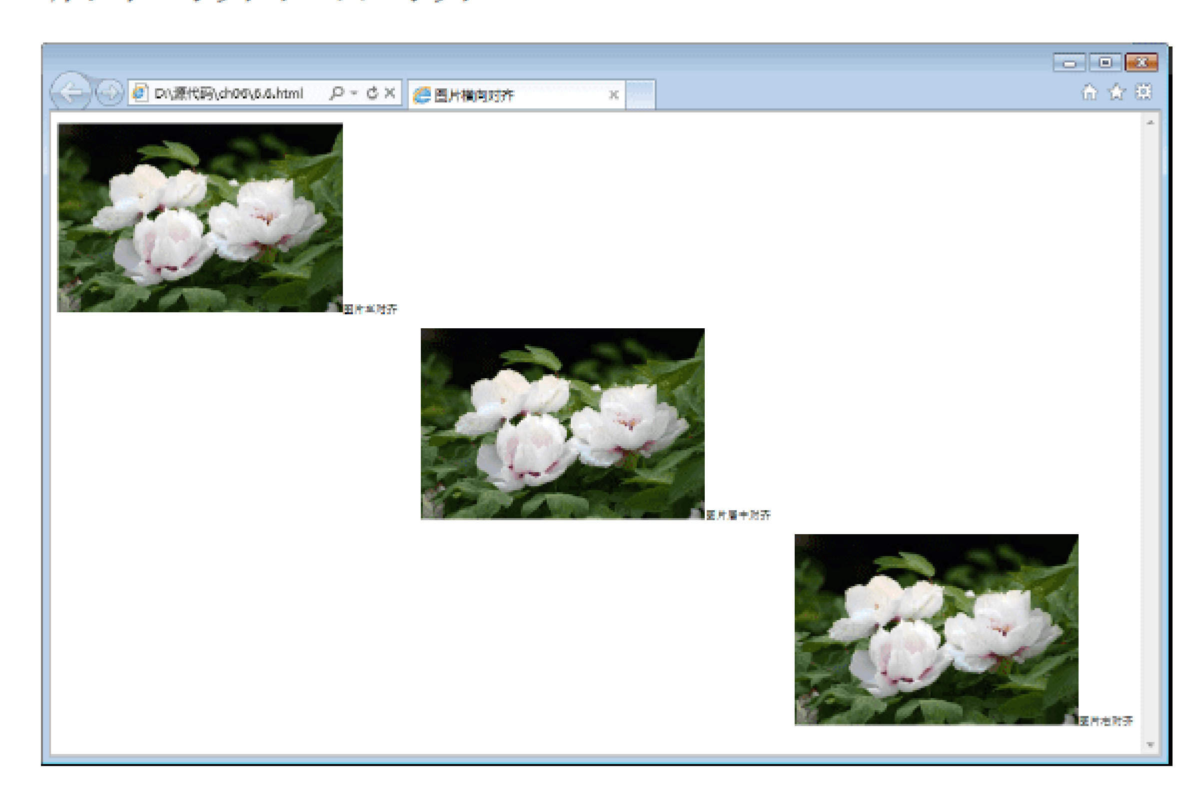


图 13-6　图片横向对齐

13.2.2　纵向对齐方式

纵向对齐就是垂直对齐，即在垂直方向上和文字搭配使用。通过对图片垂直方向上的设置，可以设定图片和文字的高度一致。在 CSS 中，对于图片的纵向设置通常使用 vertical-align 属性来定义。

vertical-align 属性设置元素的垂直对齐方式，即定义行内元素的基线相对于该元素所在行的基线的垂直对齐方式。允许指定负长度值和百分比值。这会使元素降低而不是升高。在表单元格中，这个属性会设置单元格框中的单元格内容的对齐方式。其语法格式为：

```
vertical-align : baseline |sub | super |top |text-top |middle |bottom |text-bottom |length
```

上面参数的含义如表 13-1 所示。

表 13-1　vertical-align 参数含义

参数名称	说明
baseline	支持 valign 特性的对象的内容与基线对齐
sub	垂直对齐文本的下标
super	垂直对齐文本的上标
top	将支持 valign 特性的对象的内容与对象顶端对齐
text-top	将支持 valign 特性的对象的文本与对象顶端对齐
middle	将支持 valign 特性的对象的内容与对象中部对齐
bottom	将支持 valign 特性的对象的文本与对象底端对齐
text-bottom	将支持 valign 特性的对象的文本与对象顶端对齐
length	由浮点数字和单位标识符组成的长度值或者百分数。可为负数。定义由基线算起的偏移量。基线对于数值来说为 0，对于百分数来说就是 0%

【例 13.7】（实例文件：ch13\13.7.html）

```
<html>
<head>
<title>图片纵向对齐</title>
<style>
img{
    max-width:100px;
}
</style>
</head>
<body>
<p>纵向对齐方式:baseline<img src=mudan.jpg style="vertical-align:baseline"></p>
<p>纵向对齐方式:bottom<img src=mudan.jpg style="vertical-align:bottom"></p>
<p>纵向对齐方式:middle<img src=mudan.jpg style="vertical-align:middle"></p>
<p>纵向对齐方式:sub<img src=mudan.jpg style="vertical-align:sub"></p>
<p>纵向对齐方式:super<img src=mudan.jpg style="vertical-align:super"></p>
<p>纵向对齐方式:数值定义<img src=mudan.jpg style="vertical-align:20px"></p>
</body>
</html>
```

在 IE 9.0 中浏览效果如图 13-7 所示，可以看到网页中显示了 6 张图片，垂直方向上的对齐方式分别是 baseline、bottom、middle、sub、super 和数值对齐。

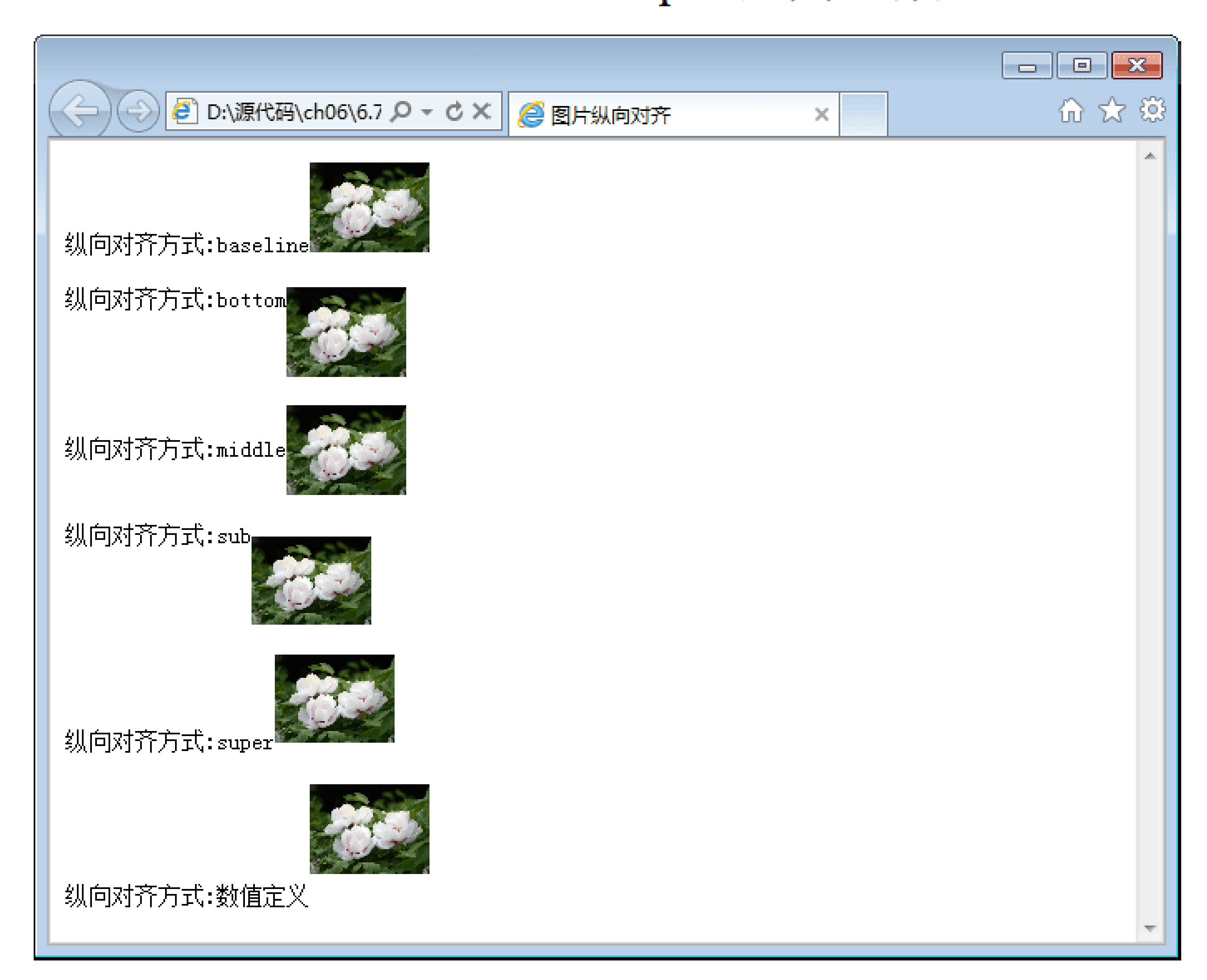


图 13-7　图片纵向对齐

13.3 图文混排

一个普通的网页最常见的方式就是图文混排，文字说明主题，图像显示情境，二者相得益彰。本节将介绍图片和文字的排版方式。

13.3.1 文字环绕

在网页中进行排版时，可以将文字设置成环绕图片的形式，即文字环绕。文字环绕应用非常广泛，如果再配合背景可以达到绚丽的效果。

在 CSS 中，可以使用 float 属性定义该效果。float 属性主要定义元素在哪个方向浮动。一般情况下，这个属性总应用于图像，使文本围绕在图像周围，有时它也可以定义其他浮动元素。浮动元素会生成一个块级框，不论它本身是何种元素。如果浮动非替换元素，则要指定一个明确的宽度；否则，它们会相当窄。

float 属性的语法格式如下所示：

```
float : none | left |right
```



其中 none 表示默认值对象不漂浮，left 表示文本流向对象的右边，right 表示文本流向对象的左边。

【例 13.8】（实例文件：ch13\13.8.html）

```
<html>
<head>
<title>文字环绕</title>
<style>
img{
    max-width:120px;
    float:left;
}
</style>
</head>
<body>
<p>
一个可爱的小家伙乘着风儿顽皮地落在了我的肩上，我低头看了看，原来是一片枫叶。
<img src="fengye.jpg">
我小心翼翼地把它捧在手中，一阵风儿吹过，叶子的几个小尖脚随风摆起，多像婴儿的小手掌啊！平滑的叶面，清晰的脉络，十分柔软细嫩。枫叶树种在秋冬的时候，体内会产生一些化学反应，让原本树叶中所含枫叶(10 张)的物质或部份组织分解之后，回收储藏在茎或根的部位，来年春天的时候可以再利用，叶绿体、叶绿素就是被分解回收的对象之一，因为叶绿素的含量较大而遮盖了其他颜色，使叶片呈绿色。因此当叶子里的叶绿素没有了的时候，其他色素的颜色彰显出来，如花青素的红色、胡萝卜素的黄色和叶黄素的黄色等。除此之外，枫叶中贮存的糖分还会分解转变成花青素，使叶片的颜色更加艳丽、火红。 枫叶没有五个"手指"
```

```
就不是枫叶,而且，枫叶的“五指”上具有锯齿，这是枫叶的特色!
    </p>
    </body>
    </html>
```

在 IE 9.0 中浏览效果如图 13-8 所示，可以看到图片被文字所环绕，并在文字的左方向显示。如果将 float 属性的值设置为 right，其图片会在文字右方显示并环绕。



图 13-8　文字环绕效果

13.3.2　设置图片与文字间距

如果需要设置图片和文字之间的距离，即文字之间存在一定间距，不是紧紧的环绕，可以使用 CSS 中的 padding 属性来设置。

padding 属性主要用来在一个声明中设置所有内边距属性，即可以设置元素所有内边距的宽度，或者设置各边上内边距的宽度。如果一个元素既有内边距又有背景，从视觉上看可能会延伸到其他行，有可能还会与其他内容重叠。元素的背景会延伸穿过内边距。不允许指定负边距值。

其语法格式如下所示：

```
padding :padding-top | padding-right | padding-bottom | padding-left
```

其参数值 padding-top 用来设置距离顶部内边距；padding-right 用来设置距离右部内边距；padding-bottom 用来设置距离底部内边距；padding-left 用来设置距离左部内边距。

【例 13.9】（实例文件：ch13\13.9.html）

```
<html>
<head>
```

```
<title>文字环绕</title>
<style>
img{
      max-width:120px;
      float:left;
      padding-top:10px;
      padding-right:50px;
      padding-bottom:10px;
}
</style>
</head>
<body>
<p>
一个可爱的小家伙乘着风儿顽皮地落在了我的肩上，我低头看了看，原来是一片枫叶。
<img src="fengye.jpg">
我小心翼翼地把它捧在手中，一阵风儿吹过，叶子的几个小尖脚随风摆起，多像婴儿的小手掌啊！平滑的叶面，清晰的脉络，十分柔软细嫩。枫叶树种在秋冬的时候，体内会产生一些化学反应，让原本树叶中所含枫叶(10 张)的物质或部份组织分解之后，回收储藏在茎或根的部位，来年春天的时候可以再利用，叶绿体、叶绿素就是被分解回收的对象之一，因为叶绿素的含量较大而遮盖了其他颜色，使叶片呈绿色。因此当叶子里的叶绿素没有了的时候，其他色素的颜色彰显出来，如花青素的红色、胡萝卜素的黄色和叶黄素的黄色等。除此之外，枫叶中贮存的糖分还会分解转变成花青素，使叶片的颜色更加艳丽、火红。枫叶没有五个"手指"就不是枫叶,而且，枫叶的“五指”上具有锯齿，这是枫叶的特色!
</p>
</body>
</html>
```

在 IE 9.0 中浏览效果如图 13-9 所示，可以看到图片被文字所环绕，并且文字和图片右边间距为 50 像素，上下各为 10 像素。



图 13-9　设置图片和文字边距

13.4　综合实例 1——一句话新闻

在各大网站中，点击率最高的就是新闻，人们每天都在不停地浏览新闻。新闻格式要求简洁明了，文字表达清楚，图文并茂。对于网页新闻排版，根据其新闻内容，可以有一句话新闻。本实例将介绍如何配合图片设计出一句话新闻的网页版面。其具体操作步骤如下所示。

01 分析需求。

在本实例中，如果要显示一句话新闻，需要包含两个部分，一个是新闻标题，一个是新闻内容，新闻内容可以使用图片和段落文字。此处可以使用 div 将两个部分分成不同的层次。实例完成后，其界面如图 13-10 所示。

02 构建 HTML 页面。

既然页面中包含了两个部分，可以使用 HTML 中的 div 标记进行层次划分。一个 div 包含整个一句话新闻，一个 div 包含标题，标题可以分为正标题和副标题；一个 div 包含图片和段落。其代码如下所示：

```
<html>
<head>
<title>时事新闻</title>
<style>
</style>
</head>
<body>
<div >
<div >
<p >英国皇家国际航展开幕</p>
<p >2011-07-17 09:38  来源：新华网</p>
</div>
<div>
<p align=center>
<img src=feiji.jpg border=1>
<p>
<p>
7 月 16 日，在英国的费尔福德，一架 A-10 攻击机进行飞行表演。为期 2 天的英国皇家国际航展当日在费尔福德空军基地开幕，这是世界上规模最大的军用飞机航空展之一。
</p>
</div>
</div>
</body>
</html>
```

在 IE 9.0 中浏览效果如图 13-11 所示，可以看到在网页中显示了段落、图片和标题。



图 13-10　一句话新闻



图 13-11　基本层次划分

03 添加 CSS 代码，修饰整体 div。

```
<style>
.da{border:#0033FF 1px solid;}
</style>
```

04 在 HTML 代码中，使用类标识符指向 da，如下所示。

```
<div class=da>
<div >
<p >英国皇家国际航展开幕</p>
<p >2011-07-17 09:38  来源：新华网</p>
</div>
<div>
<p align=center>
<img src=feiji.jpg border=1>
<p>
<p >
7 月 16 日，在英国的费尔福德，一架 A-10 攻击机进行飞行表演。为期 2 天的英国皇家国际航展当日在费尔福德空军基地开幕，这是世界上规模最大的军用飞机航空展之一。
</p>
</div>
</div>
```

05 在 IE 9.0 中浏览效果如图 13-12 所示，可以看到在网页中显示了一个边框，包围段落、图片。

06 添加 CSS 代码，修饰正标题和副标题。

```
.title{color:blue;font-size:25px;text-align:center}
.xtitle{
    text-align:center;
```

```
        font-size:13px;
        color:gray;
    }
```

07 在 HTML 代码中，引用上面两个类标识符，代码如下。

```
<div class=da>
<div>
<p class=title>英国皇家国际航展开幕</p>
<p class=xtitle>2011-07-17 09:38  来源：新华网</p>
</div>
<div>
<p align=center>
<img src=feiji.jpg border=1>
<p>
<p >
7 月 16 日，在英国的费尔福德，一架 A-10 攻击机进行飞行表演。为期 2 天的英国皇家国际航展当日在费尔福德空军基地开幕，这是世界上规模最大的军用飞机航空展之一。
</p>
</div>
</div>
```

08 在 IE 9.0 中浏览效果如图 13-13 所示，可以看到在网页中正标题和副标题都居中显示，并且正标题以蓝色显示，大小为 25 像素，副标题以灰色显示，大小为 13 像素。



图 13-12　CSS 整体修饰



图 13-13　修饰标题

09 添加 CSS 代码，修饰图片，代码如下。

```
img{
    border-top-style:solid;
    border-right-style:dashed;
    border-bottom- style:solid;
    border-left-style:dashed;
```

```
}
```

10 此处使用了标签标识符，直接作用于网页中的图片，此处就不再显示 HTML 代码了。在 IE 9.0 中浏览效果如图 13-14 所示，可以看到在网页中图片边框显示了不同样式，上下以直线显示，左右以破折线显示。

11 添加 CSS 代码，修饰段落，代码如下。

```
<p style="text-indent:10mm;font-size:15px;">
7 月 16 日，在英国的费尔福德，一架 A-10 攻击机进行飞行表演。为期 2 天的英国皇家国际航展当日在费尔福德空军基地开幕，这是世界上规模最大的军用飞机航空展之一。
</p>
```

12 在 IE 9.0 中浏览效果如图 13-15 所示，可以看到在网页中段落缩进 10mm，并且字体大小为 15 像素。



图 13-14　图片边框样式修饰



图 13-15　修饰段落

13 添加 CSS 代码，实现虚线显示。代码如下。

为了将标题和内容进行分隔，可以采用底边框实现。其代码如下所示：

```
.xiao{border-bottom:#CCCCCC 1px dashed;}
```

14 在 IE 9.0 中浏览效果如图 13-16 所示，可以看到在网页中，标题和内容被一个虚线隔开。



图 13-16　虚线分隔

13.5　综合实例 2——学校宣传单

每年暑假时，学校招生就开始了，大量的宣传单到处都是。本实例模仿一个学校宣传单，进行图文混排，从而加深前面学习的知识。具体步骤如下所示。

01 分析需求。

本实例包含两个部分，一个部分是图片信息，介绍学校场景，一个部分是段落信息，介绍学校历史和理念。这两部分都放在一个 div 中。实例完成后，效果如图 13-17 所示。



图 13-17　宣传单效果图

02 构建 HTML 网页。

创建 HTML 页面，页面中包含一个 div，div 中包含图片和两个段落信息。其代码如下所示：

```
<html >
<head>
<title>学校宣传单</title>
</head>
<body>
<div>
    <img src="xuexiao.jpg" /><p>某大学风景优美</p><p> 学校发扬“百折不挠、艰苦创业”的办学传统，坚持“质量立校、人才兴校、创新强校、文化铸校、和谐荣校”的办学理念，弘扬“爱国荣校、民主和谐、求真务实、开放创新”的精神</p>
</div>
</body>
</html>
```

在 IE 9.0 中浏览效果如图 13-18 所示，可以看到在网页中标题和内容被一个虚线隔开。

03 添加 CSS 代码，修饰 div。

```
<style>
```

```
  big{
      width:430px;
      }
</style>
```

在 HTML 代码中，将 big 引用到 div 中，代码如下所示：

```
<div class=big>
     <img src="xuexiao.jpg" /><p>某大学风景优美</p><p> 学校发扬“百折不挠、艰苦创业”的办学传统，坚持“质量立校、人才兴校、创新强校、文化铸校、和谐荣校”的办学理念，弘扬“爱国荣校、民主和谐、求真务实、开放创新”的精神</p>
</div>
```

在 IE 9.0 中浏览效果如图 13-19 所示，可以看到在网页中段落以块的形式显示。



图 13-18　HTML 页面显示

图 13-19　修饰 div 层

04 添加 CSS 代码，修饰图片。

```
img{
      width:260px;
      height:220px;
      border:#009900 2px solid;
      float:left;
      padding-right:0.5px;
}
```

在 IE 9.0 中浏览效果如图 13-20 所示，可以看到在网页中图片以指定大小显示，并且带有边框，左面进行浮动。

05 添加 CSS 代码，修饰段落。

```
p{
      font-family:"宋体";
      font-size:14px;
```

```
        line-height:20px;
    }
```

在 IE 9.0 中浏览效果如图 13-21 所示，可以看到在网页中段落以宋体显示，大小为 14 像素，行高为 20 像素。




图 13-20 修饰图片



图 13-21 修饰段落

13.6 专家解惑

1. 网页进行图文排版时，哪些是必须要做的？

在进行图文排版时，通常有下面 5 个方面需要网页设计者考虑。

（1）首行缩进：段落的开头应该空两格，html 中空格键不起作用。当然，可以用 nbsp; 来代替一个空格，但这不是理想的方式，可以用 CSS 中的首行缩进，其大小为 2em。

（2）图文混排：在 CSS 中，可以用 float，让文字在没有清理浮动的时候显示在图片以外的空白处。

（3）设置背景色：设置网页背景，增加效果。此内容会在后面介绍。

（4）文字居中：可以使用 CSS 的 text-align 属性设置文字居中。

（5）显示边框：通过 border 为图片添加一个边框。

2. 设置文字环绕时，float 元素为什么失去作用？

很多浏览器在显示未指定 width 的 float 元素时会有错误。所以不管 float 元素的内容如何，一定要为其指定 width 属性。

第 14 章

用CSS设置背景与边框

浏览页面时，首先映入眼帘的就是网页的背景色和基调，不同类型的网站有不同背景和基调。因此页面中的背景通常是网站设计时一个重要的步骤。对于单个 HTML 元素，可以通过 CSS 属性设置元素边框样式，包括宽度、显示风格和颜色等。

14.1 设置背景

背景是网页设计时的重要因素之一，一个背景优美的网页，总能吸引不少访问者。例如，喜庆类网站都是以火红背景为主题，CSS 的强大表现功能在背景方面同样发挥得淋漓尽致。

14.1.1 背景颜色

background-color 属性用于设定网页背景色，同设置前景色的 color 属性一样，该属性接受任何有效的颜色值，而对于没有设定背景色的标记，默认背景色为透明（transparent）。

其语法格式为：

```
{background-color : transparent | color}
```

关键字 transparent 是个默认值，表示透明。背景颜色 color 的设定方法可以采用英文单词、十六进制、RGB 等颜色单位。

【例 14.1】（实例文件：ch14\14.1html）

```
<html>
<head>
<title>背景色设置</title>
<head>
<body style="background-color:PaleGreen; color:Blue">
  <p>
    background-color 属性设置背景色，color 属性设置字体颜色，即前景色。
  </p>
</body>
</html>
```

在 IE 9.0 中浏览效果如图 14-1 所示，可以看到网页背景色显示浅绿色，而字体颜色为蓝色。注意，在设计网页时，背景色不要使用太艳的颜色，否则会给人一种喧宾夺主的感觉。

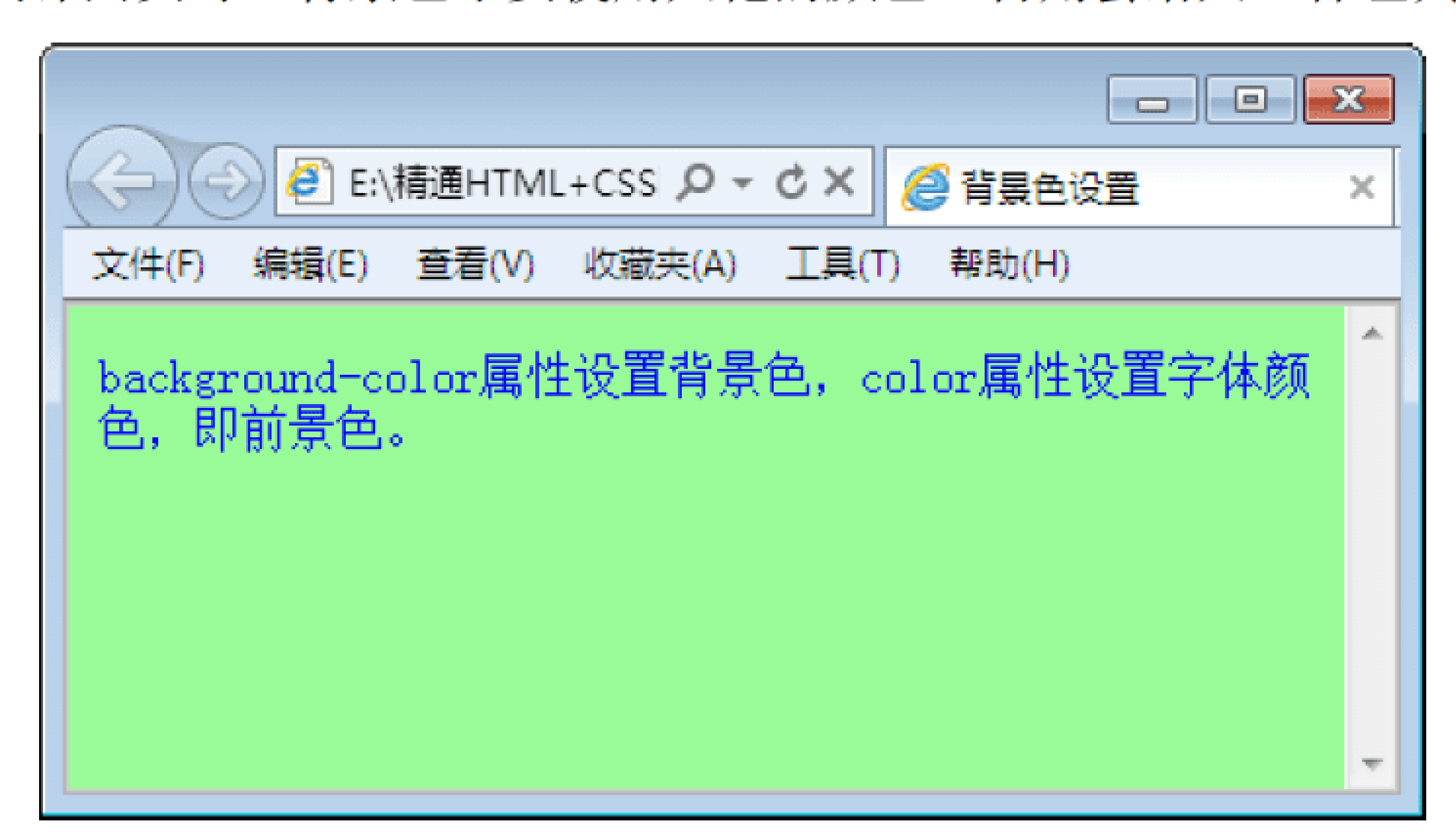


图 14-1　设置背景色

background-color 不仅可以设置整个网页的背景颜色，还可以设置指定 HTML 元素的背景色，例如设置 h1 标题的背景色、设置段落 p 的背景色。在一个网页中，可以根据需要设置不同 HTML 元素的背景色。

【例 14.2】（实例文件：ch14\14.2.html）

```
<html>
<head>
<title>背景色设置</title>
<style>
h1 {
        background-color: red;
        color: black;
        text-align:center;
}
p{
        background-color:gray;
        color:blue;
        text-indent:2em;
}
</style>
<head>
<body >
     <h1>颜色设置</h1>
   <p>
      background-color 属性设置背景色，color 属性设置字体颜色，即前景色。
   </p>
</body>
```

```
</html>
```

在 IE 9.0 中浏览效果如图 14-2 所示，可以看到网页中标题区域背景色为红色，段落区域背景色为灰色，并且分别为字体设置了不同的前景色。

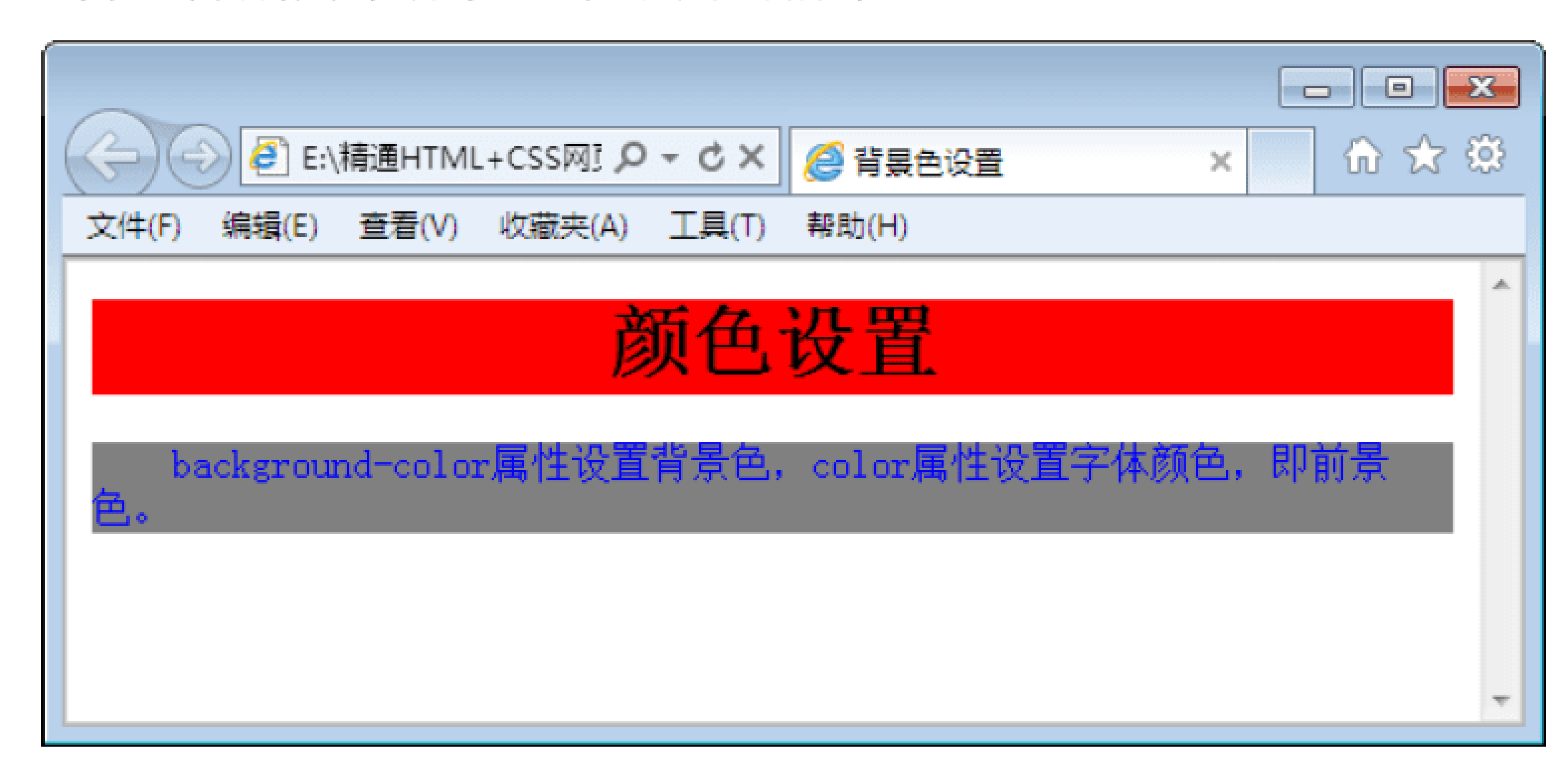


图 14-2　设置 HTML 元素背景色

14.1.2　背景图片

网页中不但可以使用背景色来填充网页背景，同样也可以使用背景图片。通过 CSS 属性可以对背景图片进行精确定位。background-image 属性用于设定标记的背景图片，通常情况下，在标记<body>中应用该属性可将图片用于整个主体中。

background-image 语法格式如下所示：

```
background-image : none | url (url)
```

其默认属性是无背景图，当需要使用背景图时可以用 url 进行导入，可以使用绝对路径，也可以使用相对路径。

【例 14.3】（实例文件：ch14\14.3.html）

```
<html>
<head>
<title>背景色设置</title>
<style>
body{
    background-image:url(xiyang.jpg)
}
</style>
<head>
<body  >
<p>夕阳无限好</p>
</body>
</html>
```

在 IE 9.0 中浏览效果如图 14-3 所示，可以看到网页中显示背景图，但如果图片大小小于整个网页大小，为了填充网页背景，图片会重复出现并铺满整个网页。

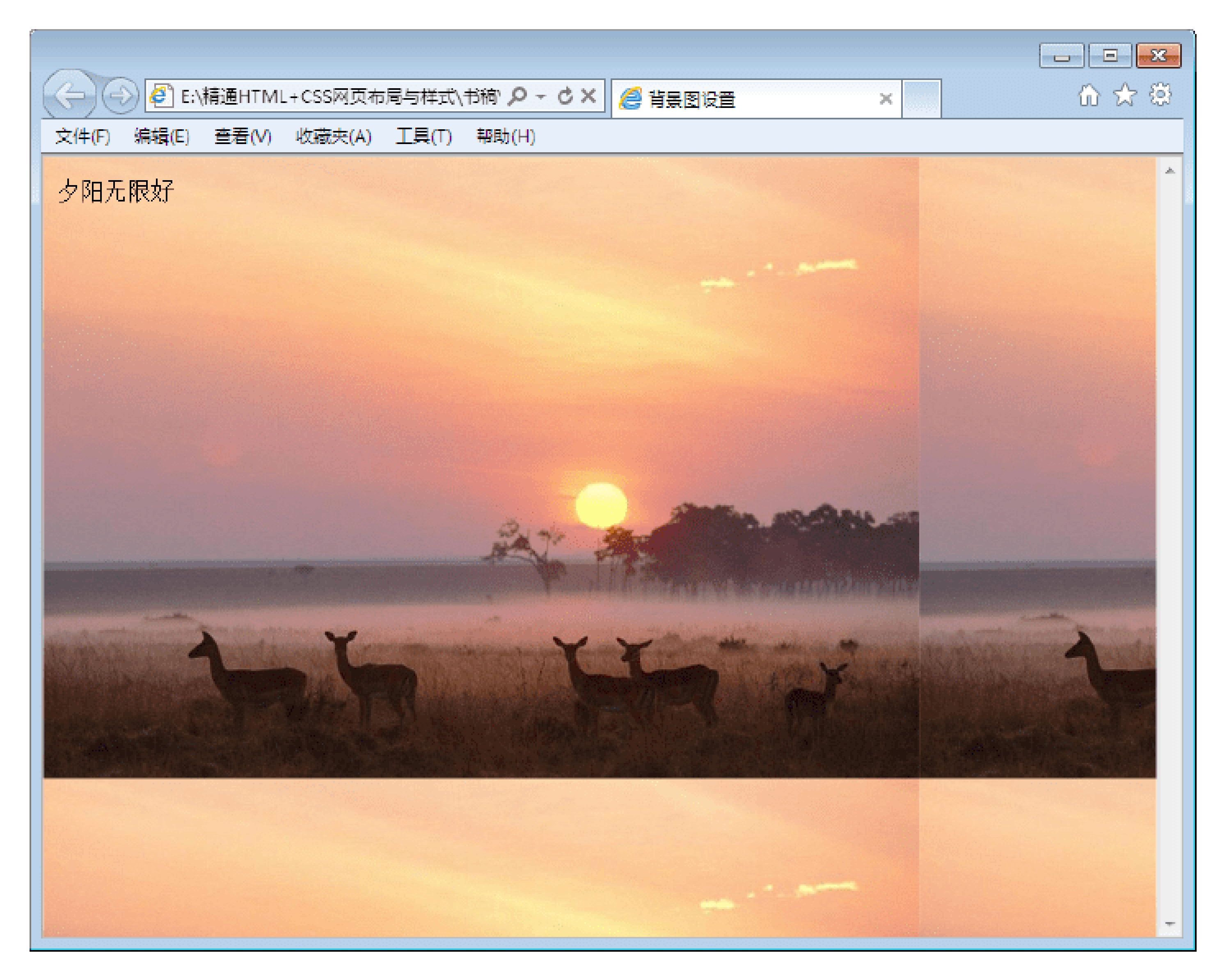


图 14-3 设置背景图片

在设定背景图片时，最好同时也设定背景色，这样当背景图片因某种原因无法正常显示时，可以使用背景色来代替。当然，如果图片正常显示，背景图片会覆盖背景色。

1. 背景图片重复

在进行网页设计时，通常都是一个网页使用一张背景图片，如果图片大小小于背景图片，会直接重复铺满整个网页，但这种方式不适用于大多数页面，在 CSS 中可以通过 background-repeat 属性设置图片的重复方式，包括水平重复、垂直重复和不重复等。

background-repeat 属性用于设定背景图片是否重复平铺。其属性值说明如表 14-1 所示。

表 14-1 background-repeat 属性及其描述

属性值	描述
repeat	背景图片水平和垂直方向都重复平铺
repeat-x	背景图片水平方向重复平铺
repeat-y	背景图片垂直方向重复平铺
no-repeat	背景图片不重复平铺

background-repeat 属性重复背景图片从元素的左上角开始平铺，直到水平、垂直或全部页面都被背景图片覆盖。

【例 14.4】（实例文件：ch14\14.4.html）

```
<html>
<head>
```

```
<title>背景图片重复</title>
<style>
body{
    background-image:url(xiyang.jpg);
    background-repeat:no-repeat;
}
</style>
<head>
<body   >
<p>夕阳无限好</p>
</body>
</html>
```

在 IE 9.0 中浏览效果如图 14-4 示，可以看到网页中显示背景图，但图片以默认大小显示，而没有对整个网页背景进行填充。这是因为代码中设置了背景图不重复平铺。

同样可以在上面代码中，设置 background-repeat 的属性值为其他值，例如可以设置值为 repeat-x，表示图片在水平方向平铺。此时，在 IE 9.0 浏览器中效果如图 14-5 所示。

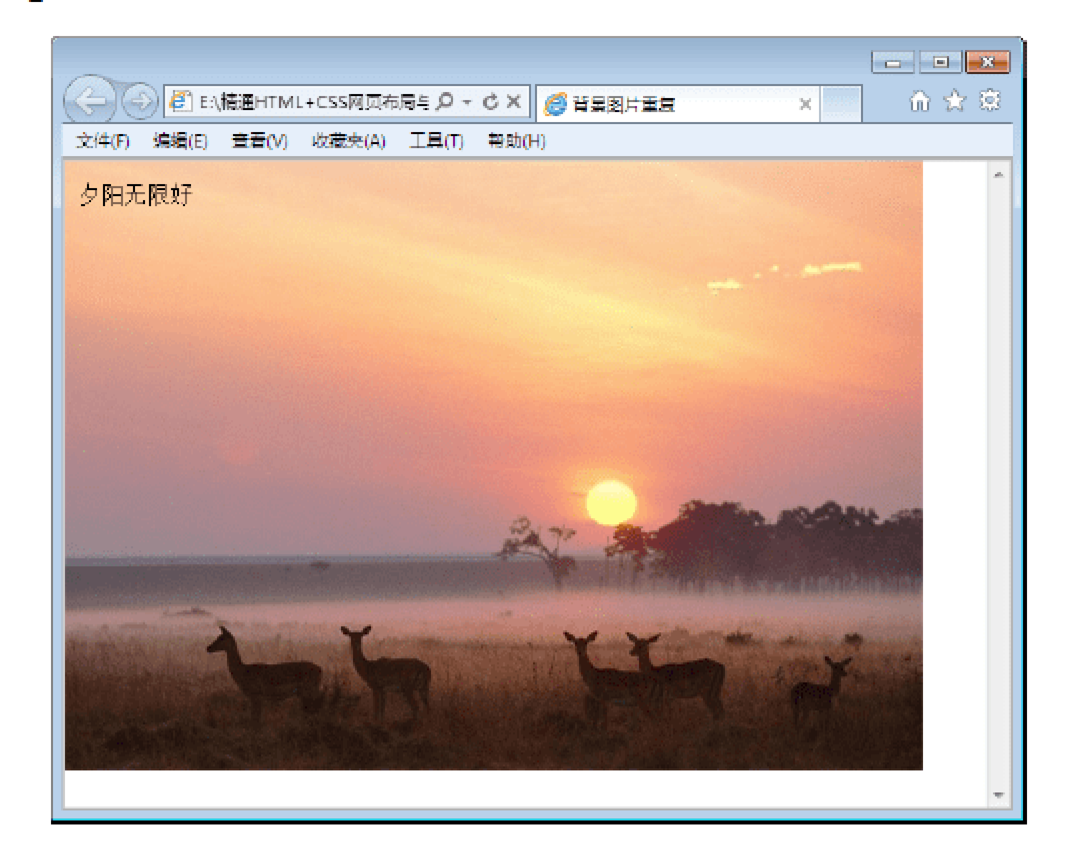


图 14-4　背景图不重复

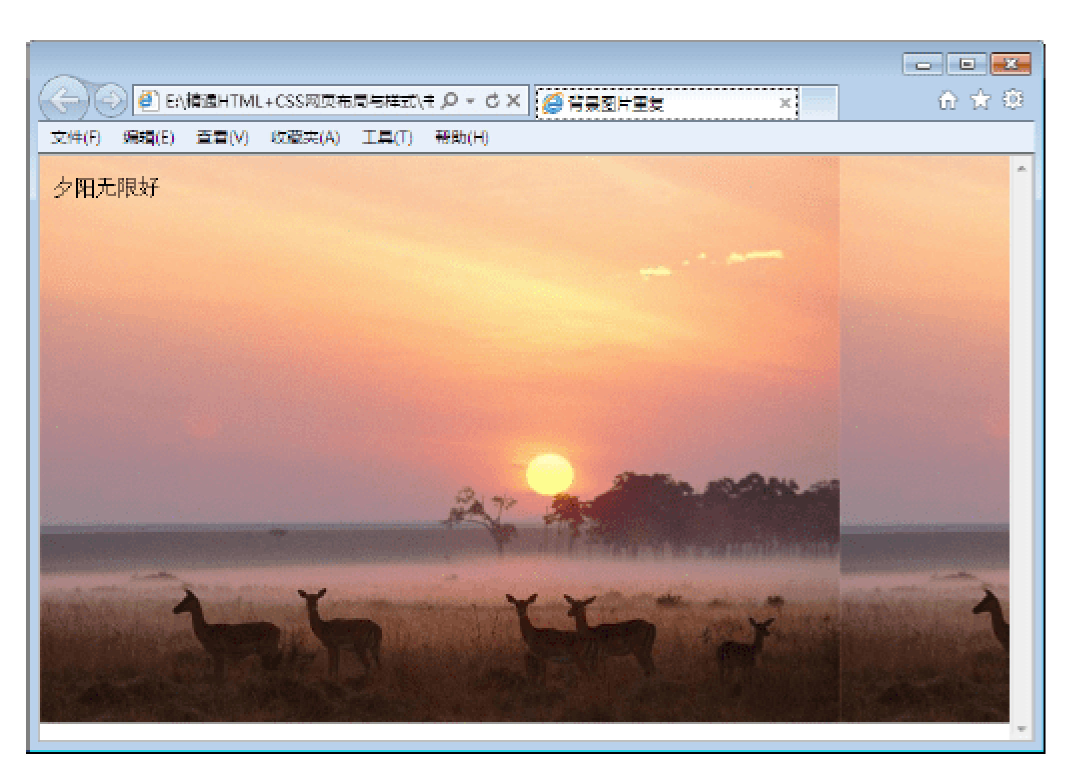


图 14-5　水平方向重复平铺

2. 背景图片显示

对于一个文本较多、一屏显示不了的页面，如果使用的背景图片不足以覆盖整个页面，而且只将背景图片应用在页面的一个位置上，那么在浏览页面时，肯定会出现看不到背景图片的情况；再者，还可能出现背景图片初始可见，而随着页面的滚动又不可见的情况。也就是说背景图片不能时刻地随着页面的滚动而显示。

要解决上述问题，用要使用 background-attachment 属性，该属性用来设定背景图片是否随文档一起滚动。该属性包含两个属性值 scroll 和 fixed，并适用于所有元素，如表 14-2 所示。

表 14-2　background-attachment 属性值及描述

属性值	描述
scroll	默认值，当页面滚动时，背景图片随页面一起滚动
fixed	背景图片固定在页面的可见区域里

使用 background-attachment 属性，可以使背景图片始终处于视野范围内，以避免出现因页面的滚动而消失的情况。

【例 14.5】（实例文件：ch14\14.5.html）

```
<html>
<head>
<title>背景显示方式</title>
<style>
body{
      background-image:url(xiyang.jpg);
      background-repeat:no-repeat;
      background-attachment:fixed;
}
p{
      text-indent:2em;
      line-height:30px;
}
h1{
        text-align:center;
}
</style>
<head>
<body    >
<h1>兰亭序</h1>
<p>
永和九年，岁在癸（guǐ）丑，暮春之初，会于会稽（kuài jī）山阴之兰亭，修禊（xì）事也。群贤毕至，少长咸集。此地有崇山峻岭，茂林修竹， 又有清流激湍（tuān），映带左右。引以为流觞（shāng）曲（ qū）水，列坐其次，虽无丝竹管弦之盛，一觞（shang）一咏，亦足以畅叙幽情。
</p>
<p>是日也，天朗气清，惠风和畅。仰观宇宙之大，俯察品类之盛，所以游目骋（chěng）怀，足以极视听之娱，信可乐也。</p>
<p> 夫人之相与，俯仰一世。或取诸怀抱，晤言一室之内；或因寄所托，放浪形骸（hái）之外。虽趣（qǔ）舍万殊，静躁不同，当其欣于所遇，暂得于己，怏然自足，不知老之将至。及其所之既倦，情随事迁，感慨系（xì）之矣。向之所欣，俯仰之间，已为陈迹，犹不能不以之兴怀。况修短随化，终期于尽。古人云："死生亦大矣。"岂不痛哉！</p>
<p>每览昔人兴感之由，若合一契，未尝不临文嗟（jiē）悼，不能喻之于怀。固知一死生为虚诞，齐彭
```

```
殇(shāng)为妄作。后之视今，亦犹今之视昔，悲夫！故列叙时人，录其所述。虽世殊事异，所以兴怀，其致一也。后之览者，亦将有感于斯文。</p>
    </body>
    </html>
```

在 IE 9.0 中浏览效果如图 14-6 所示，可以看到，当 background-attachment 属性的值为 fixed 时，背景图片的位置固定并不是相对于页面的，而相对于页面的可视范围。



图 14-6　图片显示方式

3. 背景图片的位置

我们知道，背景图片都是从设置了 background 属性的标记（例如 body 标记）的左上角开始出现，但在实际网页设计中，可以根据需要之间指定背景图片出现的位置。在 CSS 中，可以通过 background-position 属性轻松调整背景图片位置。

background-position 属性用于指定背景图片在页面中所处位置。该属性值可以分为 4 类：绝对定义位置（length）、百分比定义位置（percentage）、垂直对齐值和水平对齐值。其中垂直对齐值包括 top、center 和 bottom，水平对齐值包括 left、center 和 right，如表 14-3 所示。

表 14-3　background-position 属性值及描述

属性值	描述
length	设置图片与边距水平和垂直方向的距离长度，后跟长度单位（cm、mm、px 等）
percentage	以页面元素框的宽度或高度的百分比放置图片
top	背景图片顶部居中显示
center	背景图片居中显示
bottom	背景图片底部居中显示
left	背景图片左部居中显示
right	背景图片右部居中显示

垂直对齐值还可以与水平对齐值一起使用，从而决定图片的垂直位置和水平位置。

【例 14.6】（实例文件：ch14\14.6.html）

```
<html>
<head>
<title>背景位置设定</title>
<style>
body{
    background-image:url(xiyang.jpg);
    background-repeat:no-repeat;
    background-position:top right;
}
</style>
<head>
<body   >
</body>
</html>
```

在 IE 9.0 中浏览效果如图 14-7 所示，可以看到网页中显示的背景是从顶部和右边开始的。

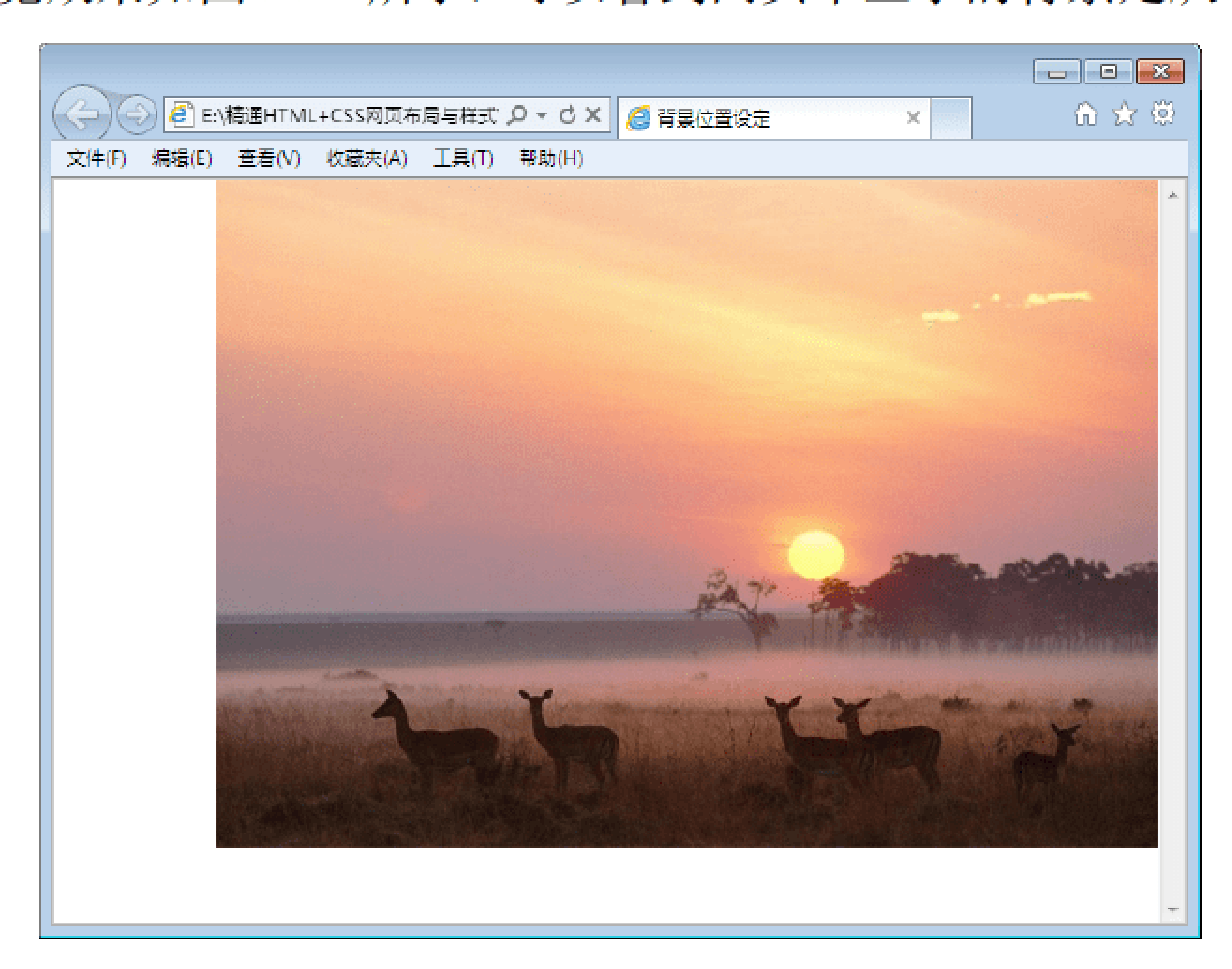


图 14-7　设置背景位置

使用垂直对齐值和水平对齐值只能格式化地放置图片，如果要在页面中自由地定义图片的位置，则需要使用确定数值或百分比。此时在上的面代码中，将

```
background-position:top right;
```

修改为

```
background-position:20px 30px
```

在 IE 9.0 中浏览效果如图 14-8 所示，可以看到网页中显示的背景从左上角开始，但并不

是（0，0）坐标位置，而是从（20，30）坐标位置开始。

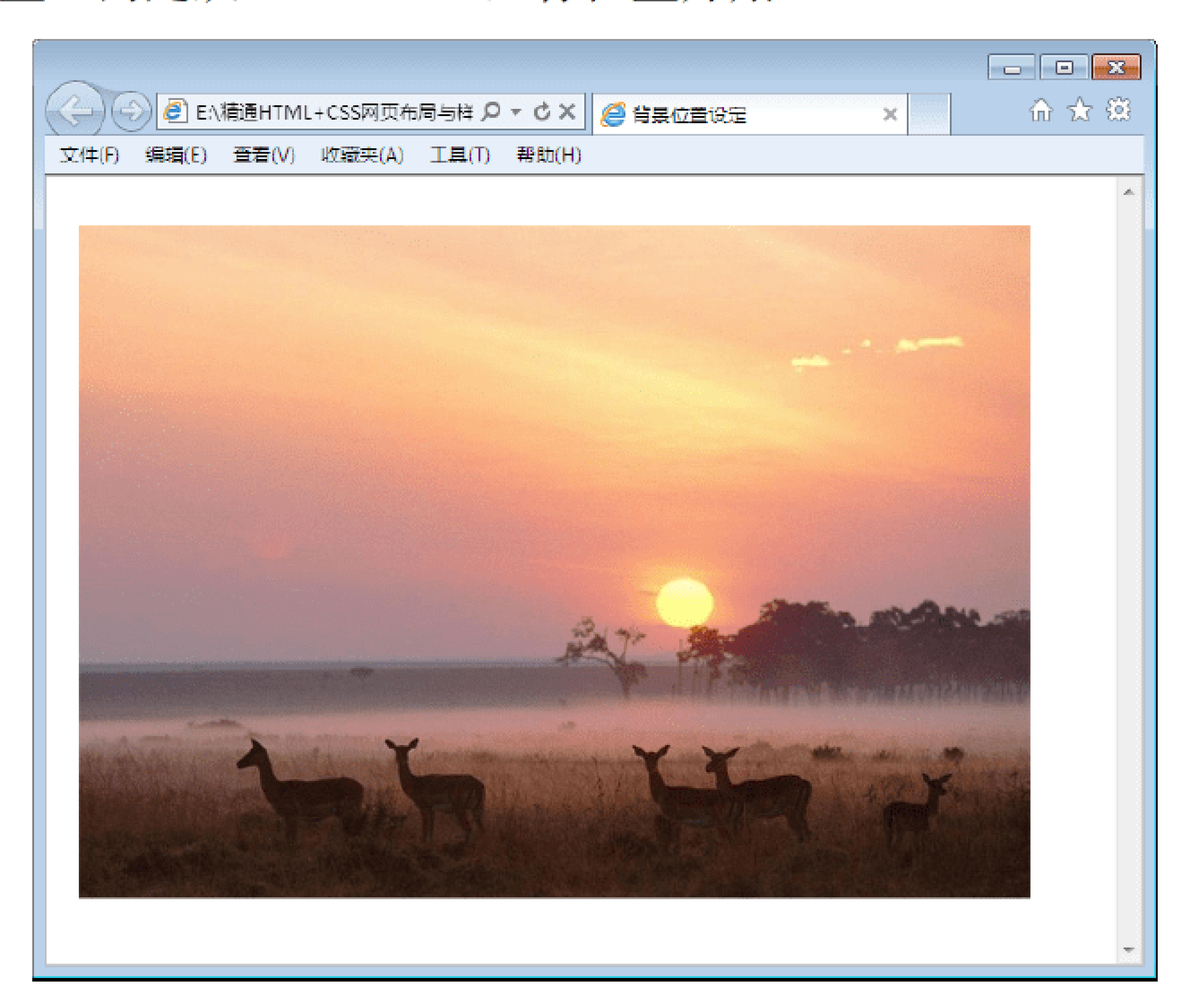


图 14-8　指定背景位置

4. 背景图片大小

在以前的网页设计中，背景图片的大小是不可以控制的，如果想要图片填充整个背景，则需要事先设计一个较大的背景图片，要么只能让背景图片以平铺的方式来填充整个页面。在 CSS 中，新增了一个 background-size 属性，用来控制背景图片大小，从而降低了网页设计的开发成本。

background-size 语法格式如下所示：

```
background-size ：[ <length> | <percentage> | auto ]{1,2} | cover | contain
```

其参数值含义如表 14-4 所示。

表 14-4　background-size 属性参数表

参数值	描述
<length>	由浮点数字和单位标识符组成的长度值。不可为负值
<percentage>	取值为 0%到 100%之间的值。不可为负值
cover	保持背景图像本身的宽高比例，将图片缩放到正好完全覆盖所定义的背景区域
contain	保持图像本身的宽高比例，将图片缩放到宽度或高度正好适应所定义的背景区域

【例 14.7】（实例文件：ch14\14.7.html）

```
<html>
<head>
<title>背景位置设定</title>
<style>
```

```
body{
    background-image:url(xiyang.jpg);
    background-repeat:no-repeat;
    background-size:cover;
}
</style>
<head>
<body   >
</body>
</html>
```

在 IE 9.0 中浏览效果如图 14-9 所示，可以看到网页中背景图片填充了整个页面。

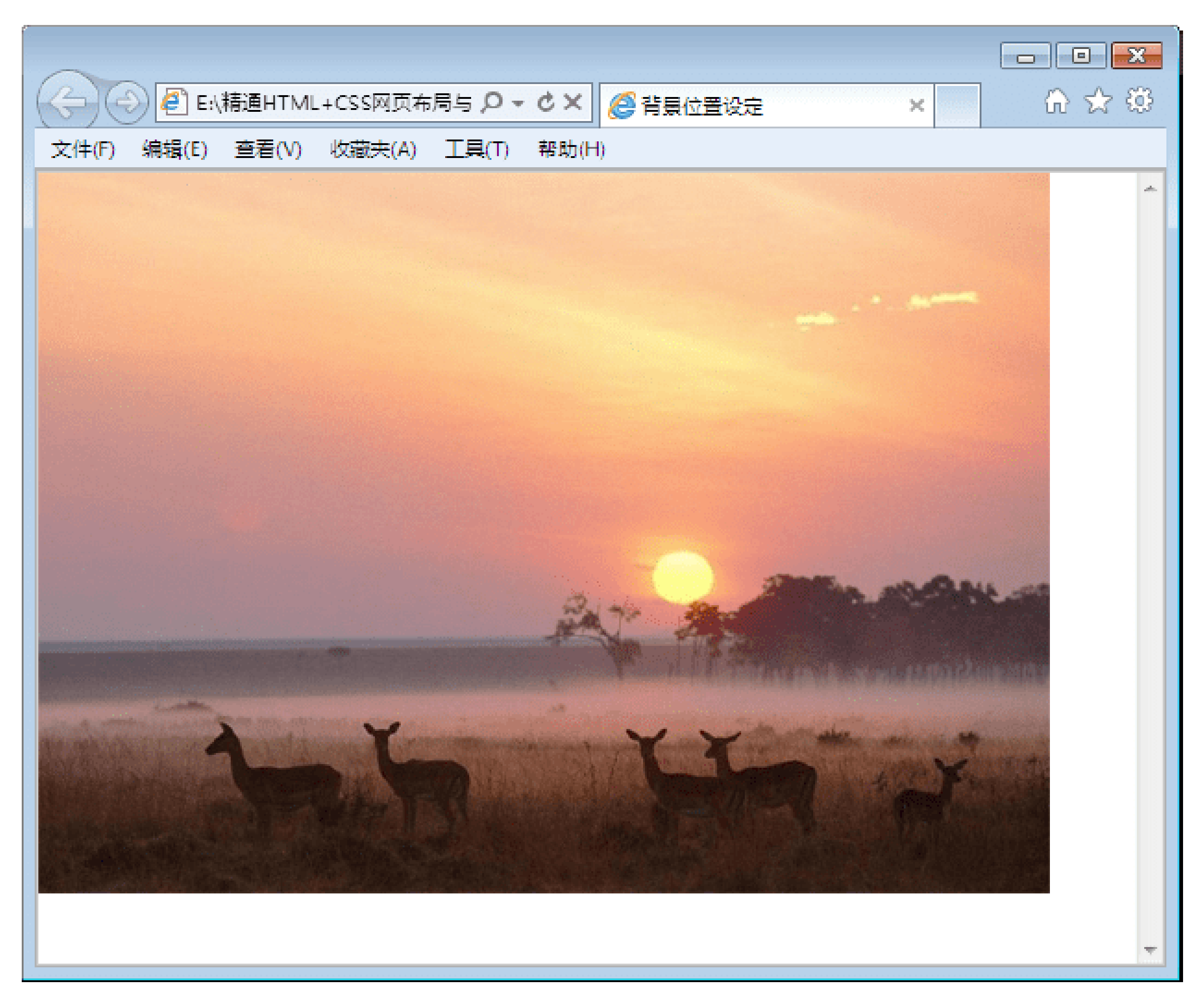


图 14-9　设定背景大小

同样也可以用像素或百分比指定背景的大小。当指定为百分比时，大小会由所在区域的宽度、高度，以及 background-origin 的位置决定。相应示例如下。

```
background-size:900 800;
```

此时 background-size 属性可以设置 1 个或 2 个值，一个值为必填，一个为选填。其中第 1 个值用于指定图片宽度，第 2 个值用于指定图片高度，如果只设定一个值，则第 2 个值默认为 auto。

14.1.3　背景复合属性

在 CSS 中，background 属性可以一起使用，即综合了以上所有与背景有关的属性（即以 back-ground-开头的属性），可以一次性地设定背景样式。格式如下：

```
background:[background-color] [background-image] [background-repeat]
```

```
[background-attachment] [background-position]
[background-size] [background-clip] [background-origin]
```

其中的属性顺序可以自由调换，并且可以选择设定。没有设定的属性，系统会自行为该属性添加默认值。例如，定义背景样式规则如下：

```
body
{
    background-color:Black;
    background-image:url(bk1.jpg);
    background-position:center;
    background-repeat:repeat-x;
    background-attachment:fixed;
    background-size:900 800;
    background-origin:padding;
    background-clip:content;
}
```

14.2　设置边框

在网页设计中，HTML 元素在网页中会占有一定的区域，例如<a>标记、<p>标记等，对于这些标记可以通过 CSS 的 width 和 height 设置其大小，并通过 border 设置其边框样式。

14.2.1　边框样式

在进行网页排版时，有时需要指定某个区域的元素，并将这些元素与其他元素区别开来，这时可以让 HTML 元素带有边框，并设置 HTML 边框样式。

Border-style 属性用于设定边框的样式，也就是风格。设定边框样式是边框最重要的部分，它主要用于为页面元素添加边框。其语法格式如下所示：

```
border-style : none | hidden | dotted | dashed | solid | double | groove | ridge | inset | outset
```

CSS 设定了 9 种边框样式，如表 14-5 所示。

表 14-5　边框样式属性及描述

属性值	描述
none	无边框，无论边框宽度设为多大
dotted	点线式边框
dashed	破折线式边框
solid	直线式边框
double	双线式边框
groove	槽线式边框

（续表）

属性值	描述
Ridge	脊线式边框
inset	内嵌效果的边框
outset	突起效果的边框

【例 14.8】（实例文件：ch14\14.8.html）

```
<html>
<head>
<title>边框设置</title>
<style>
h1 {
      border-style:dotted;
      color: black;
      text-align:center;
}
p{
      border-style:double;
      text-indent:2em;
}
</style>
<head>
<body >
    <h1>带有边框的标题</h1>
    <p>带有边框的段落</p>
</body>
</html>
```

在 IE 9.0 中浏览效果如图 14-10 示，可以看到网页中，标题 h1 显示的时候带有边框，其边框样式为点线式边框；同样段落也带有边框，其边框样式为双线式边框。

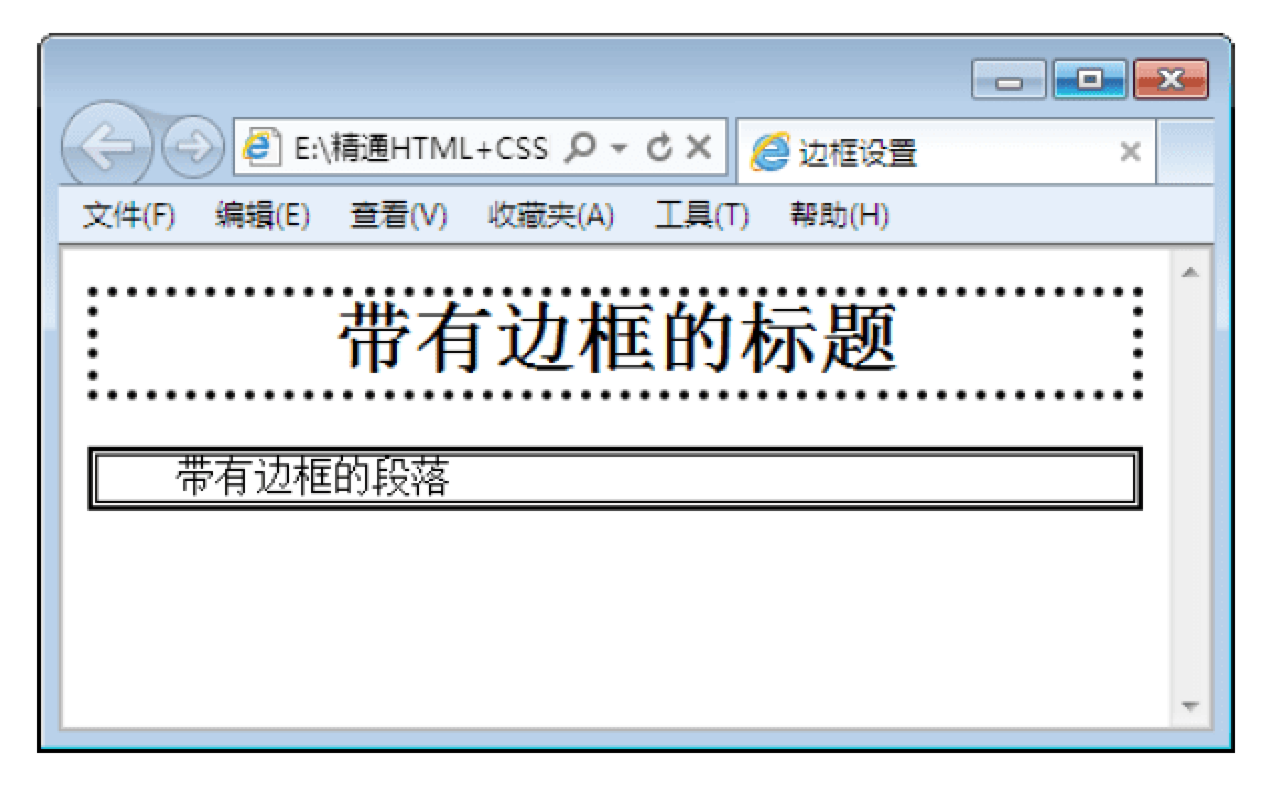


图 14-10　设置边框



在没有设定边框颜色的情况下，groove、ridge、inset 和 outset 边框默认的颜色是灰色。dotted、dashed、solid 和 double 这 4 种边框的颜色基于页面元素的 color 值。

其实，这几种边框样式还可以分别定义在一个边框中，从上边框开始按照顺时针的方向分别定义边框的上、右、下、左边框样式，从而形成多样式边框。例如，有下面一条样式规则：

```
p{border-style:dotted solid dashed groove}
```

另外，如果需要单独定义边框的一条边的样式，则可以使用表 14-6 所列的属性来定义。

表 14-6 各边样式属性及描述

属性	描述
border-top-style	设定上边框的样式
border-right-style	设定右边框的样式
border-bottom-style	设定下边框的样式
border-left-style	设定左边框的样式

14.2.2 边框颜色

在网页设计中，设计者不但可以设置边框样式，还可以设置边框颜色，从而增强边框的效果。Border-color 属性用于设定边框颜色，如果不想与页面元素的颜色相同，则可以使用该属性为边框定义其他颜色。

Border-color 属性的语法格式如下所示：

```
border-color : color
```

Color 表示指定颜色，其颜色值通过十六进制和 RGB 等方式获取。

同边框样式属性一样，border-color 属性可以为边框设定一种颜色，也可以同时设定 4 个边的颜色。

【例 14.9】（实例文件：ch14\14.9.html）

```
<html>
<head>
<title>边框颜色设置</title>
<style>
p{
    border-style:double;
    border-color:red;
    text-indent:2em;
}
</style>
<head>
```

```
<body >
        <p>边框颜色设置</p>
        <p style="border-style:solid; border-color:red blue yellow green">
    分别定义边框颜色
  </p>
</body>
</html>
```

在 IE 9.0 中浏览效果如图 14-11 所示，可以看到网页中，第一个段落边框颜色设置为红色，第二个段落边框颜色分别设置为红、蓝、黄和绿。

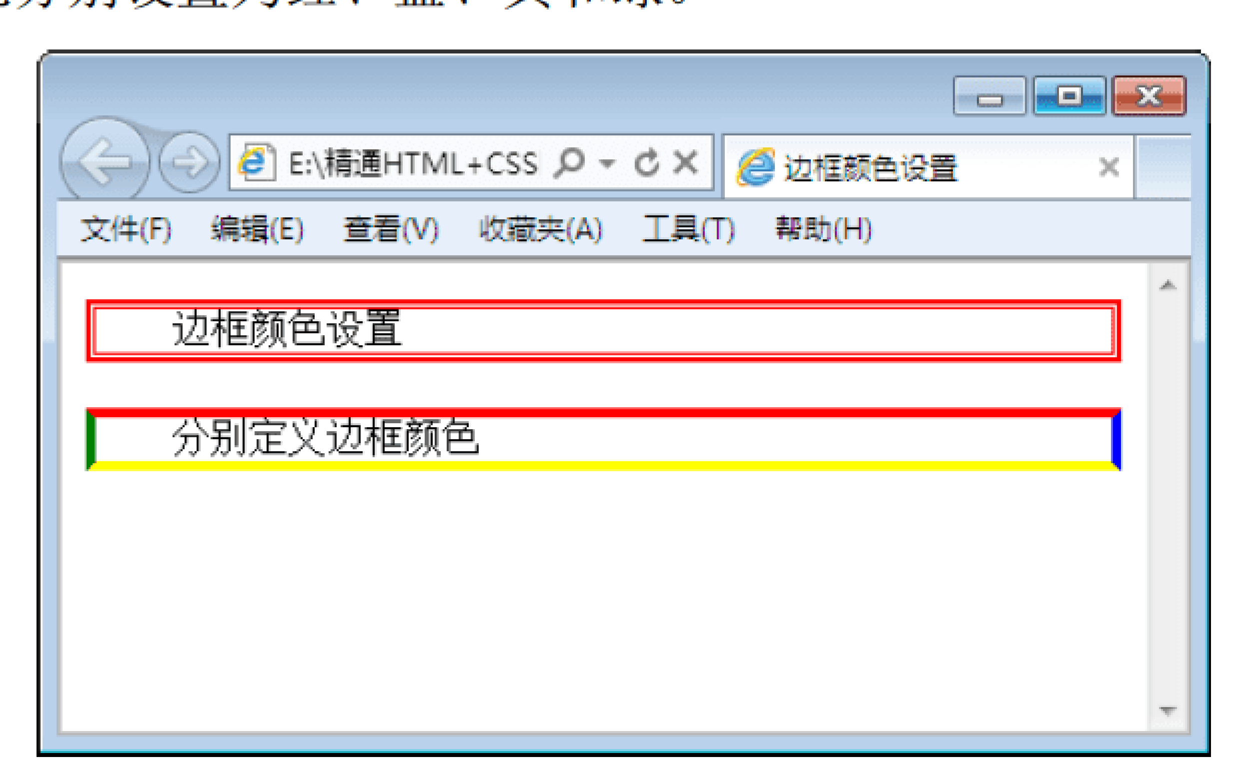


图 14-11　设置边框颜色

除了上面设置 4 个边框颜色的方法之外，还可以使用如表 14-7 列出的属性单独为相应的边框设定颜色。

表 14-7　各边颜色属性及描述

属性	描述
border-top-color	设定上边框颜色
border-right-color	设定右边框颜色
border-bottom-color	设定下边框颜色
border-left-color	设定左边框颜色

14.2.3　边框线宽

在 CSS 中，可以通过设定边框宽度，来增强边框效果。Border-width 属性就用来设定边框宽度，其语法格式如下所示：

```
border-width : medium | thin | thick | length
```

其中预设有 3 种属性值：medium、thin 和 thick，另外还可以自行设置宽度（width），如表 14-8 所示。

表 14-8　border-width 属性及描述

属性值	描述
medium	默认值，中等宽度
thin	比 medium 细
thick	比 medium 粗
length	自定义宽度

【例 14.10】（实例文件：ch14\14.10.html）

```
<html>
<head>
<title>边框宽度设置</title>
<head>
<body >
        <p style="border-style:dotted; border-width:medium;">边框颜色设置</p>
        <p style="border-style:dashed;border-width:thin;">边框颜色设置</p>
        <p style="border-style:solid; border-width:12px;">
   分别定义边框颜色
  </p>
</body>
</html>
```

在 IE 9.0 中浏览效果如图 14-12 示，可以看到网页中，3 个段落边框以不同的粗细显示。

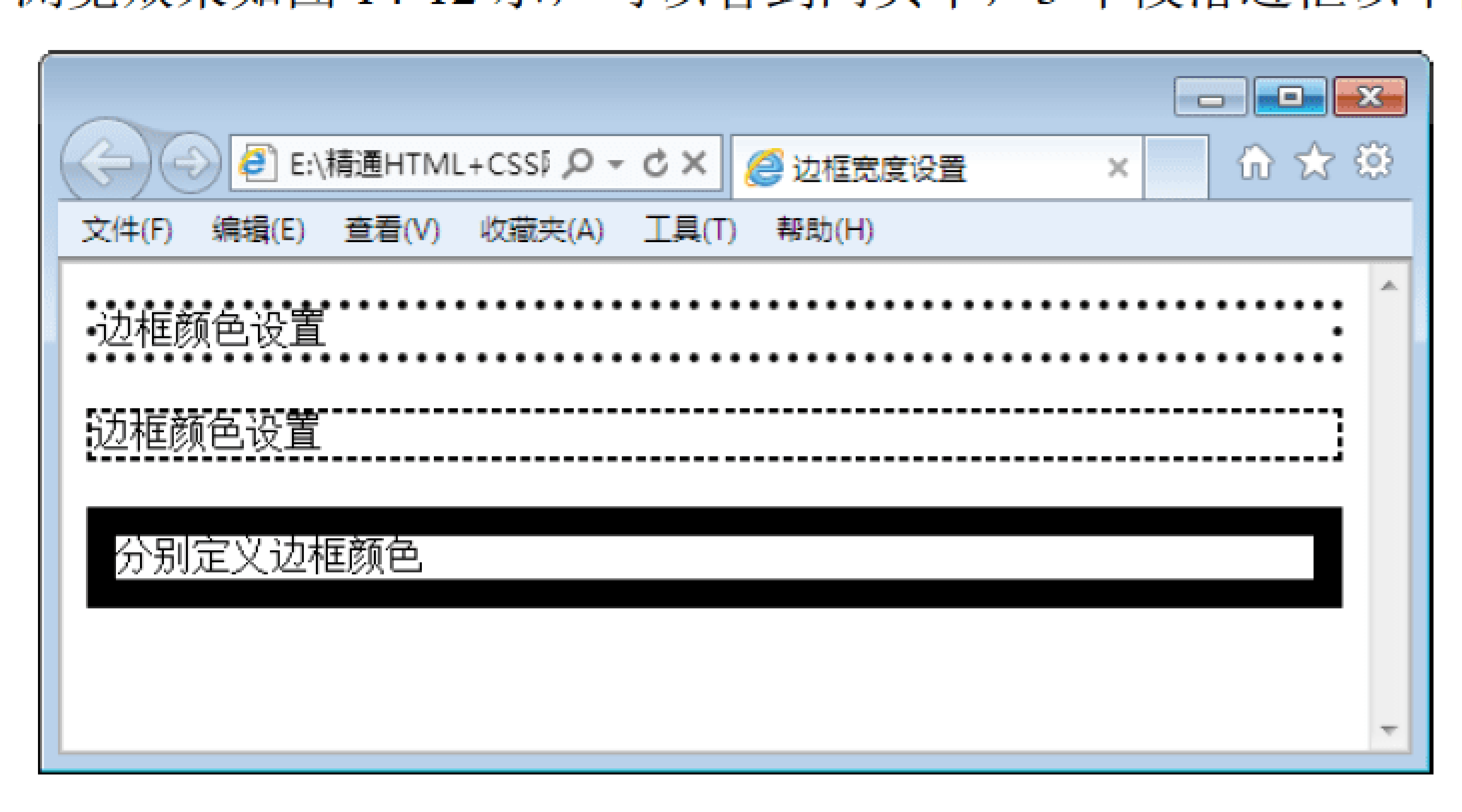


图 14-12　设置边框宽度

Border-width 属性其实是 border-top-width、border-right-width、border-bottom-width 和 border-left-width 这 4 个属性的综合属性，这 4 个属性分别用于设定上边框、右边框、下边框、左边框的宽度。

【例 14.11】（实例文件：ch14\14.11.html）

```
<html>
```

```
<head>
<title>边框宽度设置</title>
<style>
p{
    border-style:solid;
    border-color:#ff00ee;
    border-top-width:medium;
    border-right-width:thin;
    bottom-width:thick;
    border-left-width:15px;
}
</style>
<head>
<body >
     <p >边框宽度设置</p>
</body>
</html>
```

在 IE 9.0 中浏览效果如图 14-13 所示，可以看到网页中，段落的 4 个边框以不同的宽度显示。

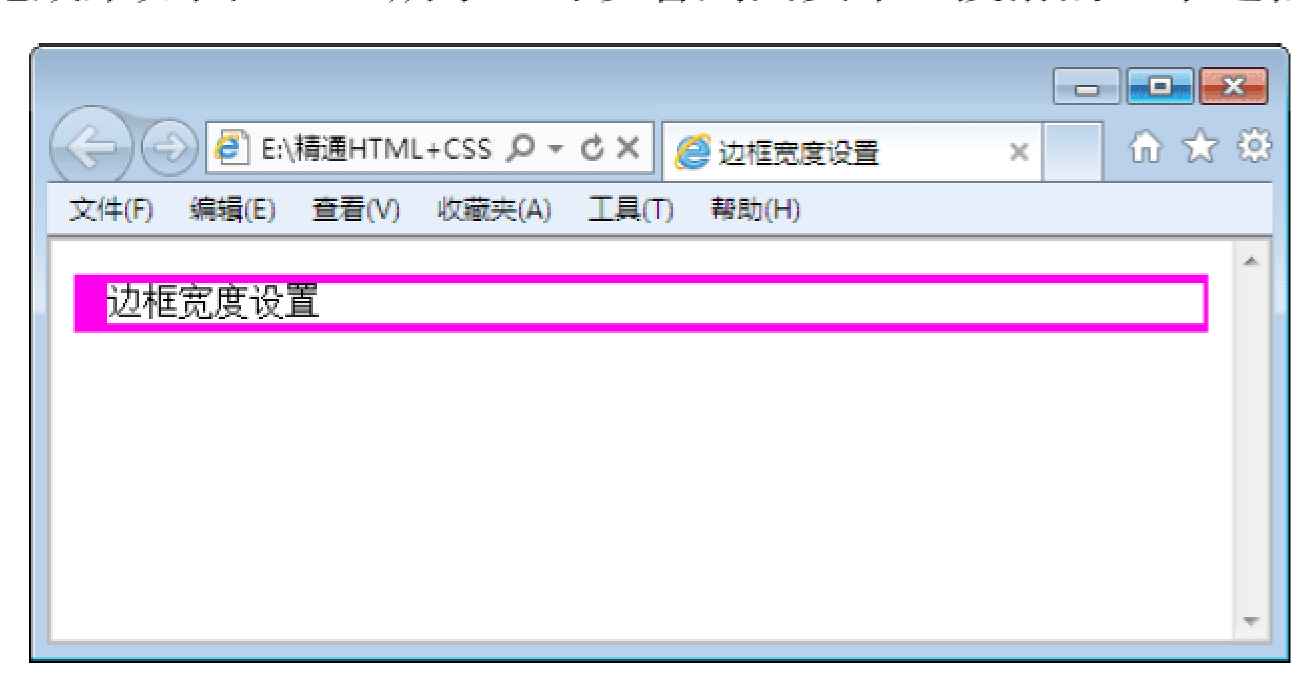


图 14-13　分别设置 4 个边框宽度

14.2.4　边框复合属性

border 属性集合了上述所介绍的 3 种属性，可以为页面元素设定边框的宽度、样式和颜色。语法格式如下所示：

```
border : border-width || border-style || border-color
```

其中，3 个属性顺序可以自由调换。

【例 14.12】（实例文件：ch14\14.12.html）

```
<html>
<head>
<title>边框复合属性设置</title>
<head>
```

```
<body >
    <p style="border:dashed  red 12px">边框复合属性设置</p>
</body>
</html>
```

在 IE 9.0 中浏览效果如图 14-14 所示，可以看到网页中，段落边框样式以破折线显示、颜色为红色、宽度为 12 像素。

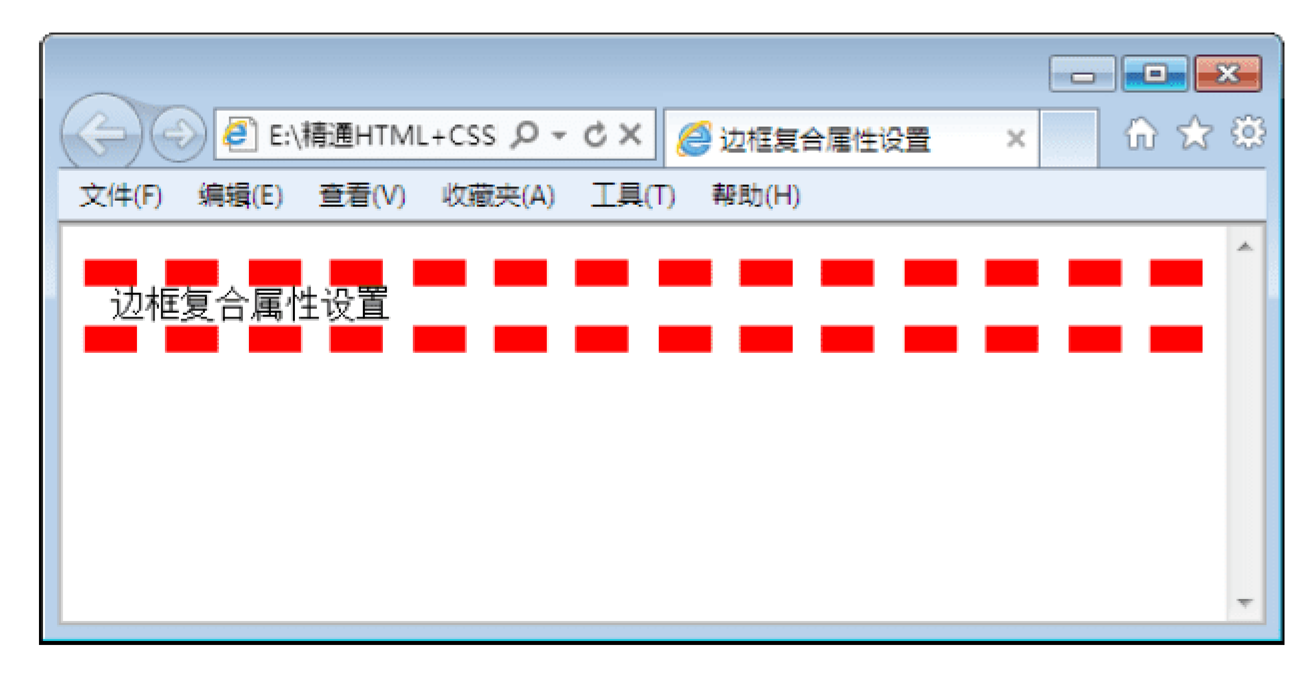


图 14-14　边框复合属性设置

14.3　综合实例——简单的公司主页

打开各种类型的商业网站，最先映入你眼帘的就是首页，也称为主页。作为一个网站的门户，一般要求主页版面整洁，美观大方。结合前面学习的背景和边框知识，我们来创建一个简单的商业网站。具体步骤如下所示。

01 分析需求。

在本实例中，主页包括 3 个部分，一部分是网站 logo，一部分是导航栏，最后一部分是主页显示内容。网站 logo 此处使用了一个背景图来代替，导航栏使用表格实现，内容列表使用无序列表实现。实例完成后，效果如图 14-15 所示。

02 构建基本 HTML。

为了划分不同的区域，HTML 页面需要包含不同的 div 层，每一层代表一个内容。一个 div 包含背景图，一个 div 包含导航栏，一个 div 包含整体内容，内容又可以划分两个不同的层。其代码如下所示：

```
<html>
<head>
<title>个人主页</title>
</head>
<body>
<center>
<div>
```

```
<div class="div1" align=center></div>
<div class=div2>
<table  width=99%><tr  align=center><td>首页</td><td>最新消息</td><td>产品展示</td><td>销售网络
</td><td>人才招聘</td><td>客户服务</td></tr></table>
</div>
<div class=div3>
<div class=div4>
<ul>最新消息
<li>公司举办 2011 金秋篮球比赛</li>
<li>消防演练大比武</li>
<li>优秀员工评比</li>
<li>公司发布招聘信息</li>
</ul>
</div>
<div class=div5>
<ul>客户案例
<li>上海电力公司</li>
<li>浙江电力公司</li>
<li>辽宁电力公司</li>
<li>河北电力公司</li>
</ul>
</div>
</div>
</div>
</center>
</body>
</html>
```

在 IE 9.0 中浏览效果如图 14-16 所示，可以看到在网页中显示了导航栏和两个列表信息。

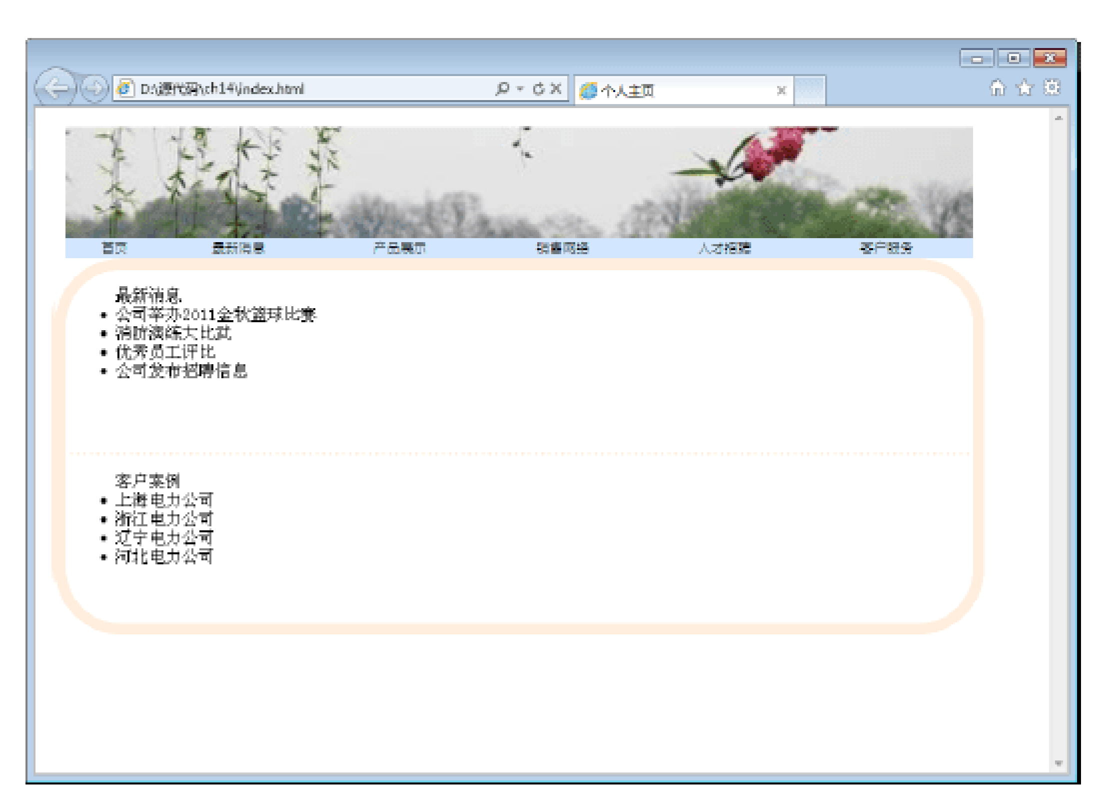


图 14-15　商业网站主页

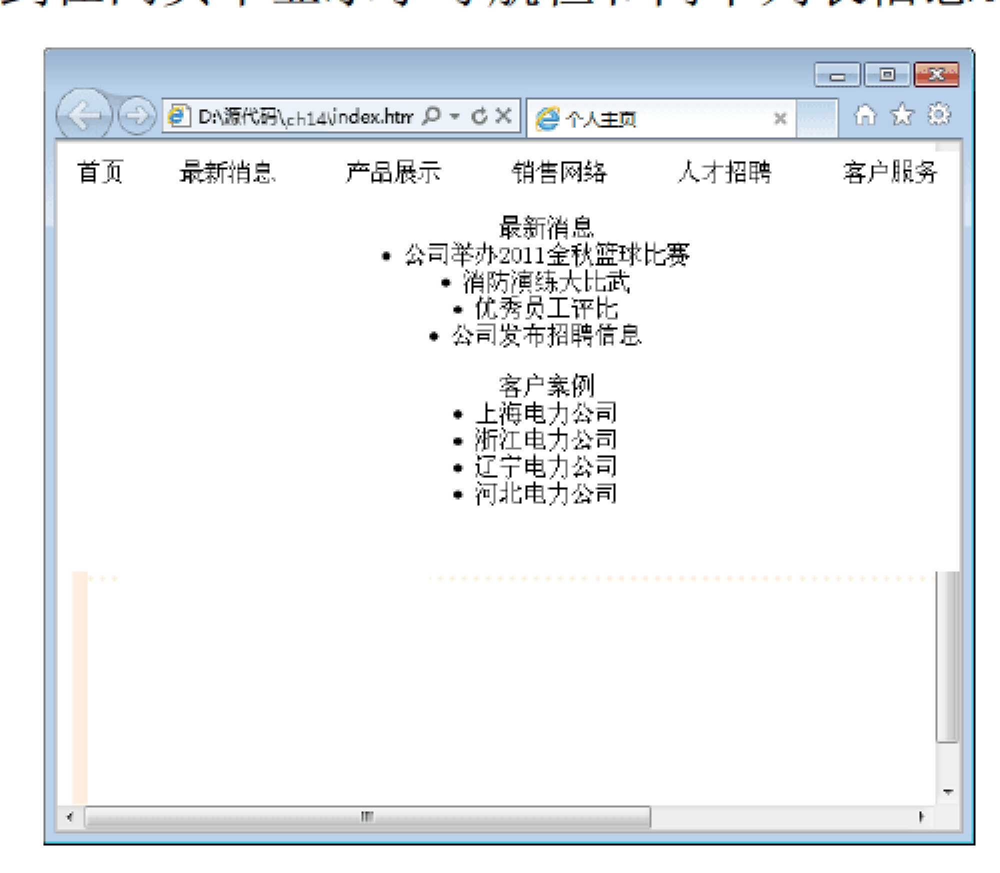


图 14-16　基本 HTML 结构

03 添加 CSS 代码，设置背景 Logo。

```
<style>
.div1{
      height:100px;
      width:820px;
      background-image:url(main.jpg);
      background-repeat:no-repeat;
      background-position:center;
      background-size:cover;

}
</style>
```

在 IE 9.0 中浏览效果如图 14-17 所示，可以看到在网页顶部显示了一个背景图，此背景覆盖整个 div 层，并不重复，并且背景图片居中显示。

04 添加 CSS 代码，设置导航栏。

```
.div2{
      width:820px;
      background-color:#d2e7ff;
}
table{
      font-size:12px;
      font-family:"幼圆";
}
```

在 IE 9.0 中浏览效果如图 14-18 所示，可以看到在网页中导航栏背景色为浅蓝色，表格中字体大小为 12 像素，字体类型是幼圆。



图 14-17　设置背景图

图 14-18　设置导航栏

05 添加 CSS 代码，设置内容样式。

```
.div3{
    width:820px;
    height:320px;
    border-style:solid;
    border-color:#ffeedd;
    border-width:10px;
}
.div4{
    width:810px;
    height:150px;
    text-align:left;
    border-bottom-width:2px;
    border-bottom-style:dotted;
    border-bottom-color:#ffeedd;
}
.div5{
    width:810px;
    height:150px;
    text-align:left;
}
```

在 IE 9.0 中浏览效果如图 14-19 所示，两个不同的内容块中间使用虚线隔开。

06 添加 CSS 代码，设置列表样式。

```
ul{
    font-size:15px;
    font-family:"楷体";
}
```

在 IE 9.0 中浏览效果如图 14-20 所示，可以看到在网页中列表字号为 15 像素，字体为楷体。

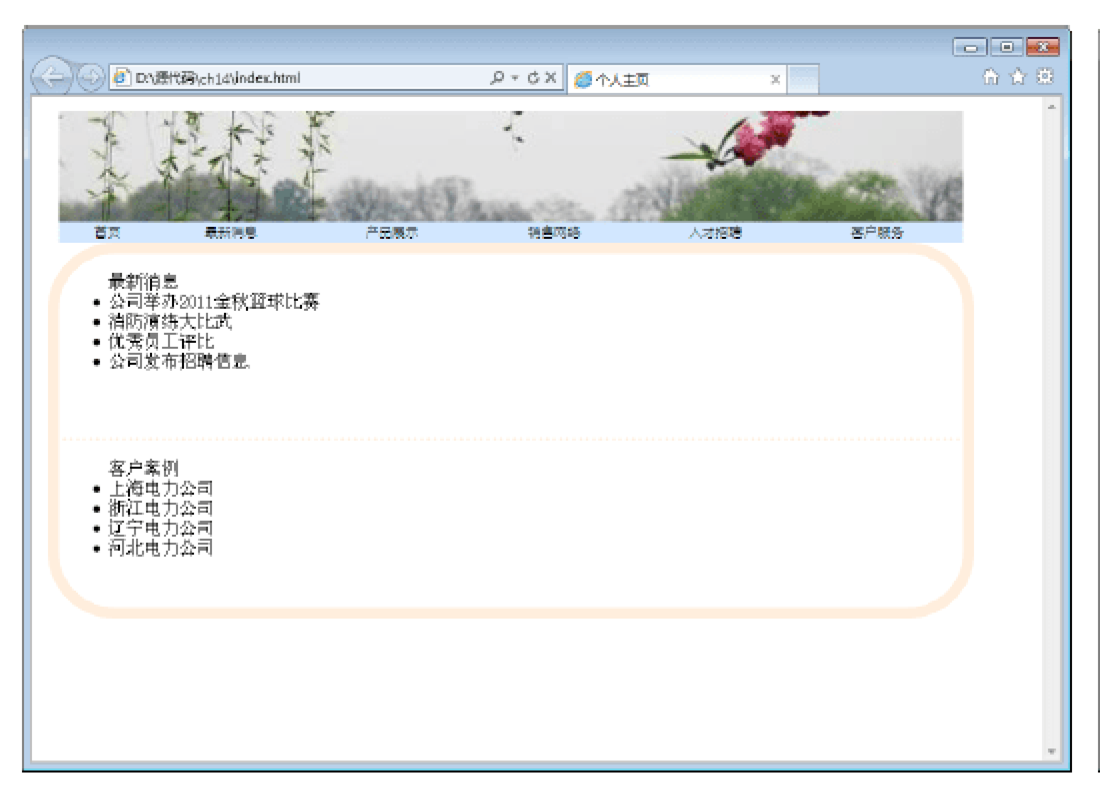




图 14-19　CSS 修饰边框



图 14-20　美化列表信息

14.4　专家解惑

1．我的背景图片不显示，是不是路径有问题？

在一般情况下，设置图片路径的代码如下。

```
background-image:url(logo.jpg);
background-image:url(../logo.jpg);
background-image:url(../images/logo.jpg);
```

对于第一种情况，要看此图片是不是与 CSS 文件在同一目录。

对于第二和第三种情况，建议尽量不要使用，因为网页文件可能存在于多级目录中，不同级目录的文件位置注定了相对路径是不一样的。而这样就让问题复杂化了，很可能图片在这个文件中可以显示正常，换了一级目录，就找不到图片的影子了。

有一种方法可以轻松解决这一问题，建立一个公共文件目录，用来存放一些公用图片文件，例如“image”，将图片文件直接存于该目录中，在 CSS 文件中，可以使用以下方式。

```
url(images/logo.jpg)
```

2．用小图片进行背景平铺好吗？

不要使用过小的图片做背景平铺。这是因为宽度和高度为 1 像素的图片平铺出一个宽度和高度为 200 像素的区域，需要复制 200*200=40, 000 次，占用资源。

3．边框样式 border:0 会占有资源吗？

推荐的写法是 border:none，虽然 border:0 只是定义边框宽度为零，但边框样式、颜色还是会被浏览器解析，占用资源。

第 15 章

用CSS美化超级链接和鼠标

超链接是网页的灵魂，各个网页都是通过超链接链接在一起的，超链接完成了页面的跳转。通过 CSS 属性定义，可以设置出具有不同外观和样式的超链接，从而增加网页样式特效。

15.1 超级链接特效

超级链接由<a>标记组成，可以是文字或图片。添加了超级链接的文字具有自己的样式，从而和其他文字区别，其中默认超级链接的样式为蓝色文字，有下划线。而通过 CSS 属性，可以修饰超级链接，从而达到美观的效果。

15.1.1 改变超级链接的基本样式

使用 HTML 标记<a>创建的超级链接非常普通，除颜色发生变化和带有下划线之外，其他的和普通文本区别不大，这种传统的超级链接样式显然无法满足广大用户的需求。此时可以通过 CSS 来增强样式效果。

对于超级链接的修饰，通常可以采用 CSS 伪类，前面已经介绍过这个概念。伪类是一种特殊的选择符，能被浏览器自动识别。其最大的用处是在不同状态下可以对超级链接定义不同的样式效果，伪类是 CSS 本身定义的一种类。

超级链接伪类的详细信息如表 15-1 所示。

表 15-1 超级链接伪类

伪类	用途
a:link	定义 a 对象在未被访问前的样式
a:hover	定义 a 对象在其鼠标悬停时的样式
a:active	定义 a 对象被用户激活时的样式（在鼠标单击与释放之间发生的事件）
a:visited	定义 a 对象在其链接地址已被访问过时样式

CSS 就是通过上面定义的 4 个超级链接伪类，设置超级链接样式。也就是说，如果要定义未被访问的超级链接的样式，可以通过 a:link 来实现，如果要设置被访问过的超级链接的样式，可以定义 a:visited 来实现。若要定义鼠标悬浮和被激活时的样式，就使用 a:hover 和 a:active 来实现。

【例 15.1】（实例文件：ch15\15.1.html）

```
<html>
<head>
<title>超级链接样式</title>
<style>
a{
    color:#545454;
    text-decoration:none;
}
a:link{
    color:#545454;
    text-decoration:none;
}
a:hover{
    color:#f60;
    text-decoration:underline;
}
a:active{
    color:#FF6633;
    text-decoration:none;
}
</style>
</head>
<body>
<center>
<a  href=#>首页</a>|<a  href=#>水月论坛</a>
<center>
</body>
</html>
```

在 IE 9.0 中浏览效果如图 15-1 所示，可以看到两个超级链接，当鼠标停留在第一个超级链接上方时，文字显示颜色为黄色，并带有下划线。另一个超级链接没有被访问，不带有下划线，颜色显示为灰色。

由此可知，伪类只提供了一种途径，用来修饰超级链接，而对超级链接真正起作用的是我们前面学习的文本、背景和边框等属性。

在网页显示的时候，有时一个超级链接并不能说明其链接背后的含义，通常还要为这个链接加上一些介绍性信息，即提示信息。此时可以通过超级链接 a 提供描述标记 title，来达到这个效果。Title 属性的值即为提示内容，当浏览器的光标停留在超级链接上时，会出现提示内容，并且不会影响页面的整洁。

【例 15.2】（实例文件：ch15\15.2.html）

```
<html>
<head>
<title>超级链接样式</title>
<style>
a{
      color:#005799;
      text-decoration:none;
}
a:link{
      color:#545454;
      text-decoration:none;
}
a:hover{
      color:#f60;
      text-decoration:underline;
}
a:active{
      color:#FF6633;
      text-decoration:none;
}
</style>
</head>
<body>
<a href="" title="这是一个优秀的团队">了解我们</a>
</body>
</html>
```

在 IE 9.0 中浏览效果如图 15-2 所示，可以看到当鼠标停留超级链接上方时，文字显示颜色为黄色，带有下划线，并且显示提示信息“这是一个优秀的团队”。

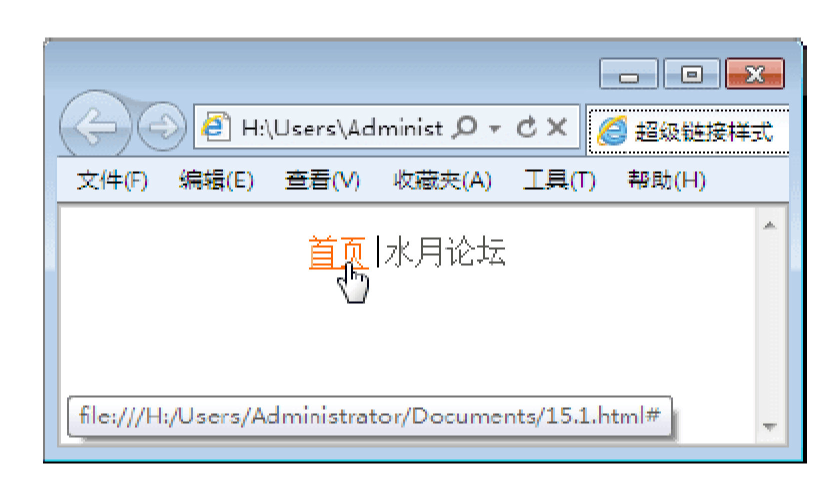


图 15-1 使用伪类修饰超级链接

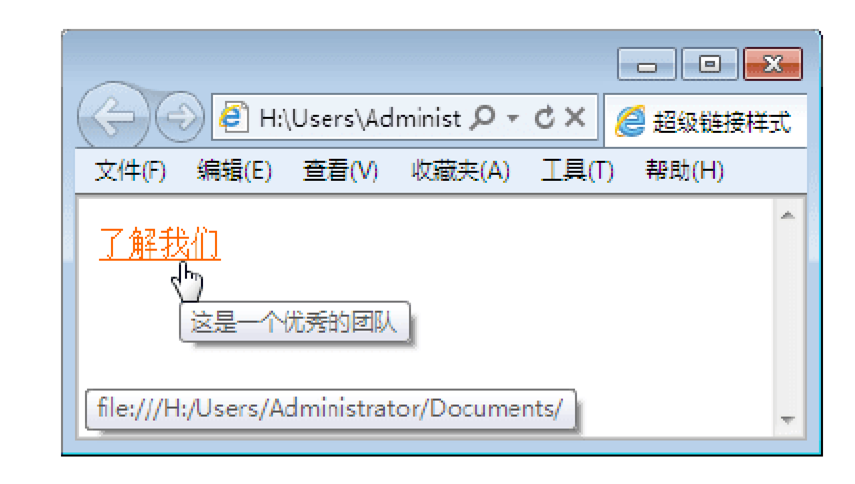


图 15-2 超级链接提示信息

15.1.2 超级链接背景图设置

一个普通的超级链接，要么是以文本显示，要么是以图片显示，显示方式很单一。可以将图片作为背景图添加到超级链接里，这样链接呈现更加精美的效果。超级链接如果要添加背景

图片，通常使用 background-image 属性来完成。

【例 15.3】（实例文件：ch15\15.3.html）

```
<html>
<head>
<title>超级链接样式</title>
<style>
a{
    background-image:url(17.jpg);
    width:90px;
    height:30px;
    color:#005799;
    text-decoration:none;
}
a:hover{
    background-image:url(115.jpg);
    color:#006600;
    text-decoration:underline;
}
</style>
</head>
<body>
<a href="#">炼器</a>
<a href="#">炼丹</a>
<a href="#">炼符</a>
</body>
</html>
```

在 IE 9.0 中浏览效果如图 15-3 所示，可以看到页面中显示了 3 个超级链接，当鼠标停留在一个超级链接上时，其背景图就会显示蓝色并且文字带有下划线，而当鼠标不在超级链接上时，背景图显示浅蓝色，并且文字不带有下划线。

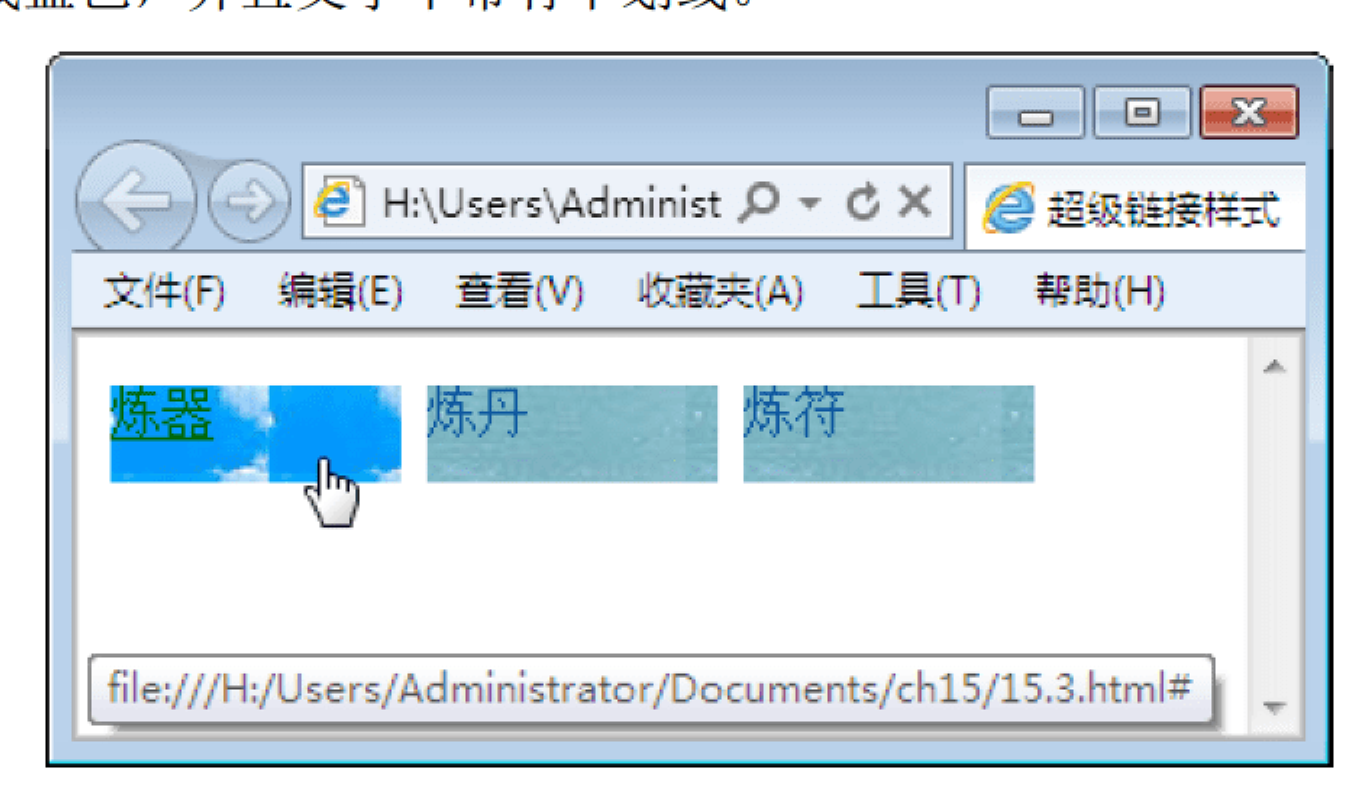


图 15-3　图片超级链接

在上面的代码中，使用 background-image 属性引入背景图，使用 text-decoration 属性设置超级链接是否具有下划线。

15.1.3 超级链接按钮效果

有时为了增强超级链接的效果，会将超级链接模拟成表单按钮，即当鼠标指针移动到一个超级链接上的时候，超级链接的文本或图片就会像被按下一样，有一种凹陷的效果。其实现方式通常是利用 CSS 中的 a:hover 属性，当鼠标经过链接时，会将链接向下、向右各移一个像素，这时候显示效果就好像按钮被按下。

【例 15.4】（实例文件：ch15\15.4.html）

```
<html>
<head>
<title>超级链接样式</title>
<style>
a{
	font-family:"幼圆";
	font-size:2em;
	text-align:center;
	margin:3px;
}
a:link,a:visited{
	color:#ac2300;
	padding:4px 10px 4px 10px;
	background-color:#ccd8db;
	text-decoration:none;
	border-top:1px solid #EEEEEE;
	border-left:1px solid #EEEEEE;
	border-bottom:1px solid #717171;
	border-right:1px solid #717171;
}
a:hover{
	color:#821818;
	padding:5px 8px 3px 12px;
	background-color:#e2c4c9;
	border-top:1px solid #717171;
	border-left:1px solid #717171;
	border-bottom:1px solid #EEEEEE;
	border-right:1px solid #EEEEEE;
}
</style>
</head>
```

```
<body>
<a href="#">首页</a>
<a href="#">水月日记</a>
<a href="#">水月学习</a>
<a href="#">闲言碎语</a>
<a href="#">生活趣事</a>
</body>
</html>
```

在 IE 9.0 中浏览效果如图 15-4 所示，可以看到页面中显示了 5 个超级链接，当鼠标停留在一个超级链接上时，其背景色显示黄色并具有凹陷的感觉，而当鼠标不在超级链接上时，背景显示浅灰色。



图 15-4 超级按钮效果

上面的 CSS 代码中，需要对<a>标记进行整体控制，同时加入了 CSS 的 2 个伪类属性。对于普通超链接和单击过的超链接采用同样的样式，并且边框的样式模拟按钮效果。而对于鼠标指针经过时的超级链接，相应地改变文本颜色、背景色、位置和边框，从而模拟按下的效果。

15.2 鼠标特效

对于经常操作计算机的人来说，当鼠标移动到不同地方，或执行不同操作时，鼠标样式是不同的。例如，当需要伸缩窗口时，将鼠标放置在窗口边沿处，鼠标会变成双向箭头状；当系统繁忙时，鼠标会变成漏斗状。如果要在网页中实现这种效果，可以通过 CSS 属性定义实现。

15.2.1 使用 CSS 控制鼠标箭头

CSS 不仅能够准确地控制及美化页面，而且还能定义鼠标指针样式。当鼠标移动到不同的 HTML 元素对象上面时，鼠标会以不同形状或图像显示。CSS 通过改变 cursor 属性（鼠标指针属性）来实现对鼠标样式的控制。

Cursor 属性包含 17 个属性值，对应鼠标的 17 个样式，而且还能够通过 url 链接地址自定义鼠标指针，如表 15-2 所示。

表 15-2　鼠标样式

属性值	说明
auto	自动，按照默认状态自行改变
crosshair	精确定位十字
default	默认鼠标指针
hand	手形
move	移动
help	帮助
wait	等待
text	文本
n-resize	箭头朝上双向
s-resize	箭头朝下双向
w-resize	箭头朝左双向
e-resize	箭头朝右双向
ne-resize	箭头右上双向
se-resize	箭头右下双向
nw-resize	箭头左上双向
sw-resize	箭头左下双向
pointer	指示
url (*url*)	自定义鼠标指针

【例 15.5】（实例文件：ch15\15.5.html）

```
<html>
<head>
<title>鼠标特效</title>
</head>
<body>
  <h2>CSS 控制鼠标箭头</h2>
  <div style="font-size:10pt;color:DarkBlue">
    <p style="cursor:hand">手形</p>
    <p style="cursor:move">移动</p>
    <p style="cursor:help">帮助</p>
    <p style="cursor:n-resize">箭头朝上双向</p>
    <p style="cursor:ne-resize">箭头右上双向</p>
     <p style="cursor:wait">等待</p>
  </div>
</body>
</html>
```

在 IE 9.0 中浏览效果如图 15-5 所示，可以看到多个鼠标样式提示信息，当鼠标放到“帮助”文字上时，鼠标会以问号“？”显示，从而起到提示作用。读者可以将鼠标放在不同的文字上，查看不同的鼠标样式。



图 15-5　鼠标样式



需要重点说明的是自定义鼠标样式，为使页面更加美观，许多网站使用了各种样式的鼠标指针来代替传统鼠标指针。

15.2.2　鼠标变幻超链接

知道了如何控制鼠标样式，就可以轻松制作出鼠标指针样式变幻的超级链接效果，即将鼠标放到超级链接上，可以看到超级链接颜色、背景图片发生变化，并且鼠标样式也会发生变化。

【例 15.6】（实例文件：ch15\15.6.html）

```
<html>
<head>
<title>鼠标手势</title>
<style>
a{
    display:block;
    background-image:url(03.jpg);
    background-repeat:no-repeat;
    width:100px;
    height:30px;
    line-height:30px;
    text-align:center;
    color:#FFFFFF;
    text-decoration:none;
}
a:hover{
    background-image:url(115.jpg);
```

```
        color:#FF0000;
        text-decoration:none;
    }
    .help{
        cursor:help;
    }
    .text{cursor:text;}
    </style>
    </head>
    <body>
    <a href="#" class="help">帮助我们</a>
    <a href="#" class="text">招聘信息</a>
    </body>
    </html>
```

在 IE 9.0 中浏览效果如图 15-6 所示，可以看到当鼠标放到“帮助我们”上时，其鼠标样式以问号显示，字体颜色显示为红色，背景色为蓝天白云。当鼠标不在“帮助我们”上时，背景图片为绿色，字体颜色为白色。

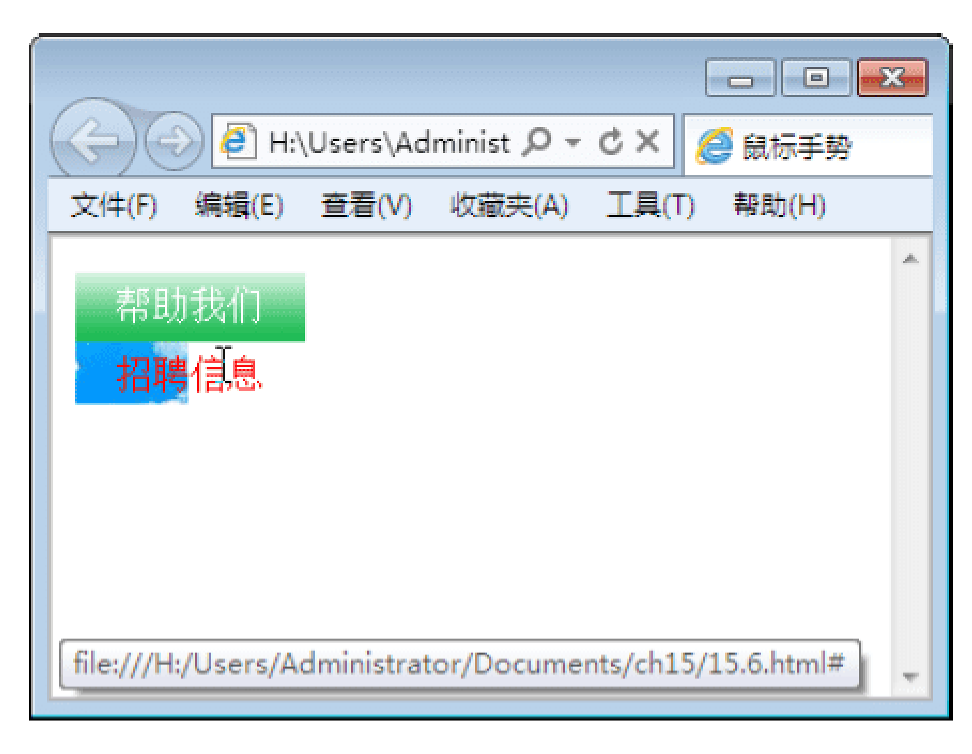


图 15-6　鼠标变幻效果

15.2.3　页面滚动条

当一个网页内容较多的时候，浏览器窗口不能在一屏内完全显示，就会给浏览者提供滚动条，方便读者浏览相关内容。对于 IE 浏览器，可以单独设置滚动条样式，从而满足网站整体样式设计。

滚动条主要由 3dlight、highlight、face、arrow、shadow 和 darkshadow 几个部分组成。其具体含义如表 15-3 所示。

表 15-3　滚动条属性设置

滚动条属性	CSS 版本	兼容性	描述
scrollbar-highlight-color	IE 专有属性	IE5.5+	设置或检索滚动条 3D 界面的亮边（ThreedHighlight）颜色

（续表）

滚动条属性	CSS 版本	兼容性	描述
scrollbar-3d-light-color	IE 专有属性	IE5.5+	设置或检索滚动条亮边框颜色
scrollbar-face-color	IE 专有属性	IE5.5+	设置或检索滚动条 3D 表面（ThreedFace）的颜色
scrollbar-arrow-color	IE 专有属性	IE5.5+	设置或检索滚动条方向箭头的颜色
scrollbar-shadow-color	IE 专有属性	IE5.5+	设置或检索滚动条 3D 界面的暗边（ThreedShadow）颜色
scrollbar-dark-shadow-color	IE 专有属性	IE5.5+	设置或检索滚动条暗边框（ThreedDarkShadow）颜色
scrollbar-base-color	IE 专有属性	IE5.5+	设置或检索滚动条基准颜色。其他界面颜色将据此自动调整

【例 15.7】（实例文件：ch15\15.7.html）

```
<html>
<head>
<title>设置滚动条</title>
<style>
body{
      overFlow-x:hidden;
      overFlow-y:scroll;
      scrollBar-face-color:green;
      scrollBar-hightlight-color:red;
      scrollBar-3dlight-color:orange;
      scrollBar-darkshadow-color:blue;
      scrollBar-shadow-color:yellow;
      scrollBar-arrow-color:purple;
      scrollBar-track-color:black;
      scrollBar-base-color:pink;
  }
p{
      text-indent:2em;
}
  </style>
</head>
<body>
<h1 align=center>岳阳楼记</h1>
<p>
庆历四年春，滕子京谪守巴陵郡。越明年，政通人和，百废具兴。乃重修岳阳楼，增其旧制，刻唐贤今人诗赋于其上。属（zhǔ）予作文以记之。
</p>
          <p>
```

```
    予观夫巴陵胜状，在洞庭一湖。衔远山，吞长江，浩浩汤汤（shāngshāng），横无际涯。朝晖夕阴，气象万千。此则岳阳楼之大观也，前人之述备矣。然则北通巫峡，南极潇湘，迁客骚人，多会于此，览物之情，得无异乎？
    </p><p>
    若夫霪雨霏霏，连月不开，阴风怒号，浊浪排空。日星隐曜，山岳潜形。商旅不行，樯倾楫摧。薄暮冥冥，虎啸猿啼。登斯楼也，则有去国怀乡，忧谗畏讥，满目萧然，感极而悲者矣。
    </p><p>
    至若春和景明，波澜不惊，上下天光，一碧万顷。沙鸥翔集，锦鳞游泳。岸芷汀（tīng）兰，郁郁青青。而或长烟一空，皓月千里，浮光跃金，静影沉璧，渔歌互答，此乐何极！登斯楼也，则有心旷神怡，宠辱偕忘，把酒临风，其喜洋洋者矣。 </p><p>
    嗟夫！予尝求古仁人之心，或异二者之为。何哉？不以物喜，不以己悲；居庙堂之高，则忧其民，处江湖之远，则忧其君。是进亦忧，退亦忧。然则何时而乐耶？其必曰“先天下之忧而忧，后天下之乐而乐”乎？噫！微斯人，吾谁与归？
    </p><p>
    时六年九月十五日。
    </p>
    </body>
    <html>
```

在 IE 9.0 中浏览效果如图 15-7 所示，可以看到页面显示了一个绿色滚动条，滚动条边框显示黄色，箭头显示为紫色。

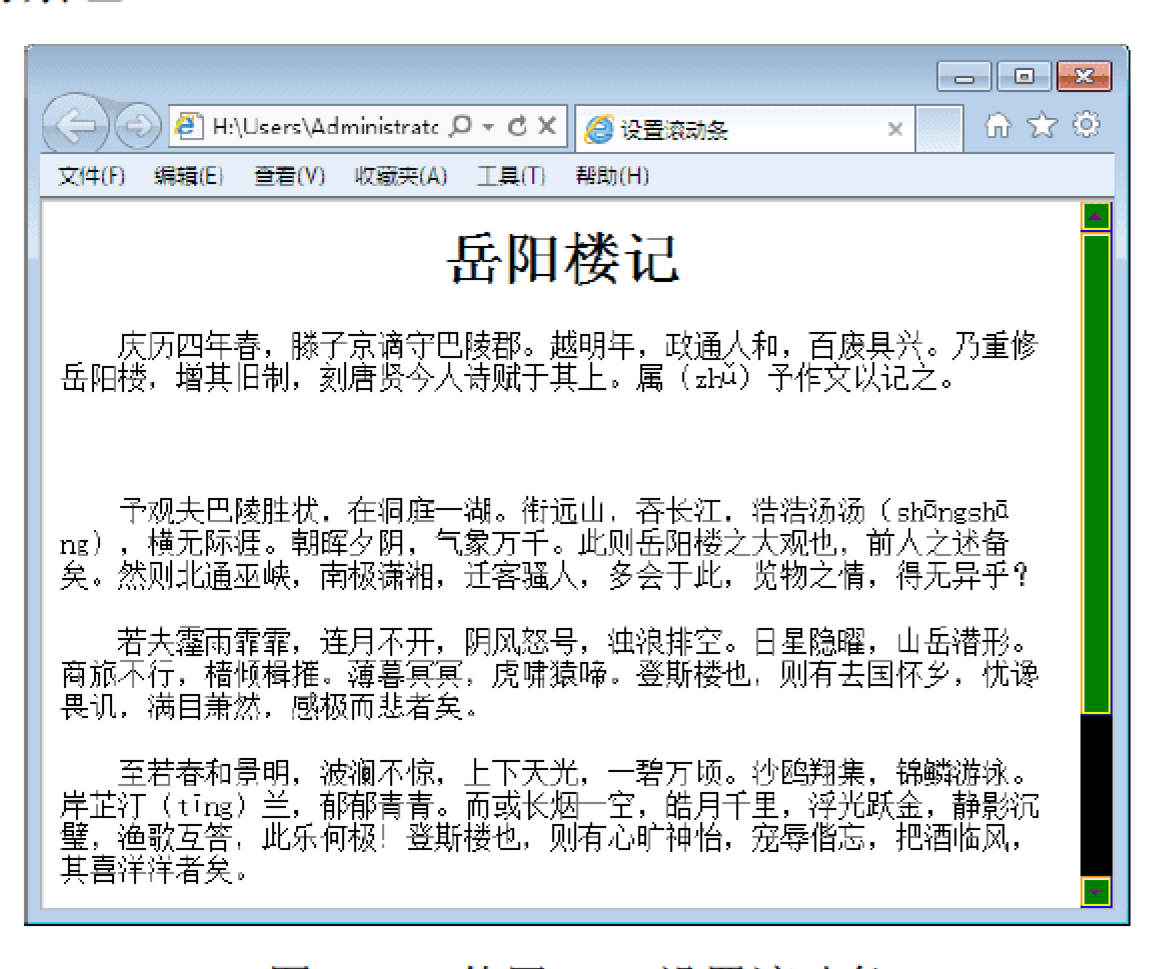


图 15-7　使用 CSS 设置滚动条

15.3　综合实例 1——制作实用导航栏

网站每个页面中，基本都有一个导航栏，作为浏览者跳转的入口。导航栏一般由超级链接创建，导航栏的样式可以采用 CSS 来设置。导航栏样式变化基于文字、背景图片和边框的变化。

结合前面学习的知识，创建一个实用工具栏。具体步骤如下所示。

01 分析需求。

通常需要在导航栏上创建一些超级链接，然后对这些超级链接进行修饰。超级链接可以横排，也可以竖排。链接上可以导入背景图片，文字上也可以添加下划线。实例完成后，其效果如图 15-8 所示。

02 构建 HTML，创建超级链接。

```
<html >
<head>
<title>制作导航栏</title>
</head>
<body>
<a href="#">最新消息</a>
<a href="#">产品展示</a>
<a href="#">客户中心</a>
<a href="#">联系我们</a>
</body>
</html>
```

在 IE 9.0 中浏览效果如图 15-9 所示，可以看到页面中创建了 4 个超级链接，其排列方式为横排，颜色为蓝色，文字带有下划线。

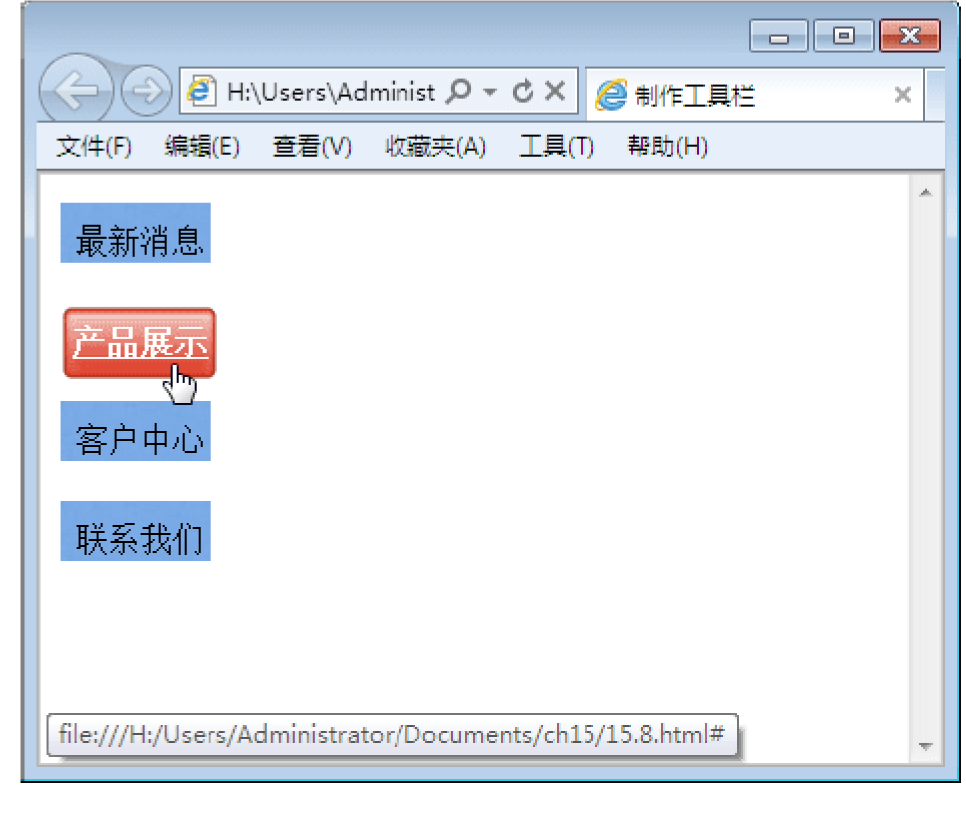


图 15-8　导航栏效果

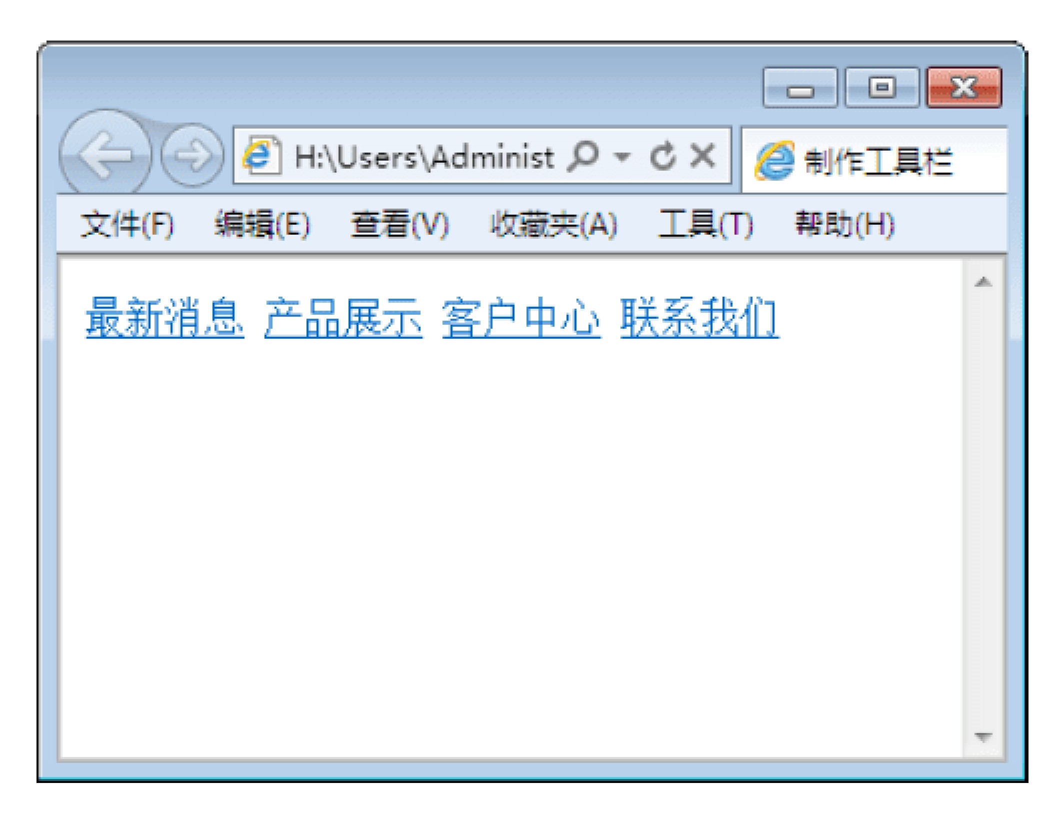


图 15-9　创建超级链接

03 添加 CSS 代码，修饰超级链接的样式。

```
<style type="text/css">
<!--
a, a:visited {
    display: block;
    font-size:16px;
    height: 50px;
```

```
        width: 80px;
        text-align: center;
        line-height: 40px;
        color: #000000;
        background-image: url(20.jpg);
        background-repeat: no-repeat;
        text-decoration: none;
}
-->
</style>
```

在 IE 9.0 中浏览效果如图 15-10 所示，可以看到页面中 4 个超级链接排列方式变为竖排，并且每个链接都导入了一个背景图片，超级链接高度为 50 像素，宽度为 80 像素，字体颜色为黑色，文字不带有下划线。

04 添加 CSS 代码，修饰链接悬浮样式。

```
a:hover {
        font-weight: bolder;
        color: #FFFFFF;
        text-decoration: underline;
        background-image: url(hover.gif);
}
```

在 IE 9.0 中浏览效果如图 15-11 所示，可以看到当鼠标放到导航栏上的一个超级链接上时，其背景图片发生了变化，文字带有下划线。

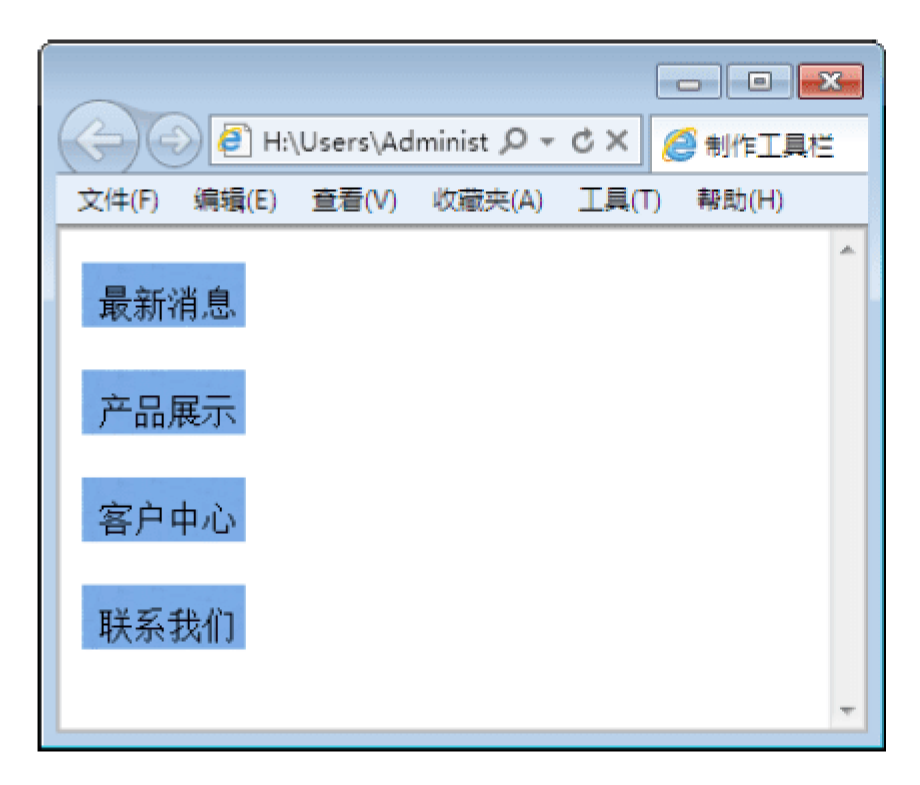


图 15-10　设置链接基本样式

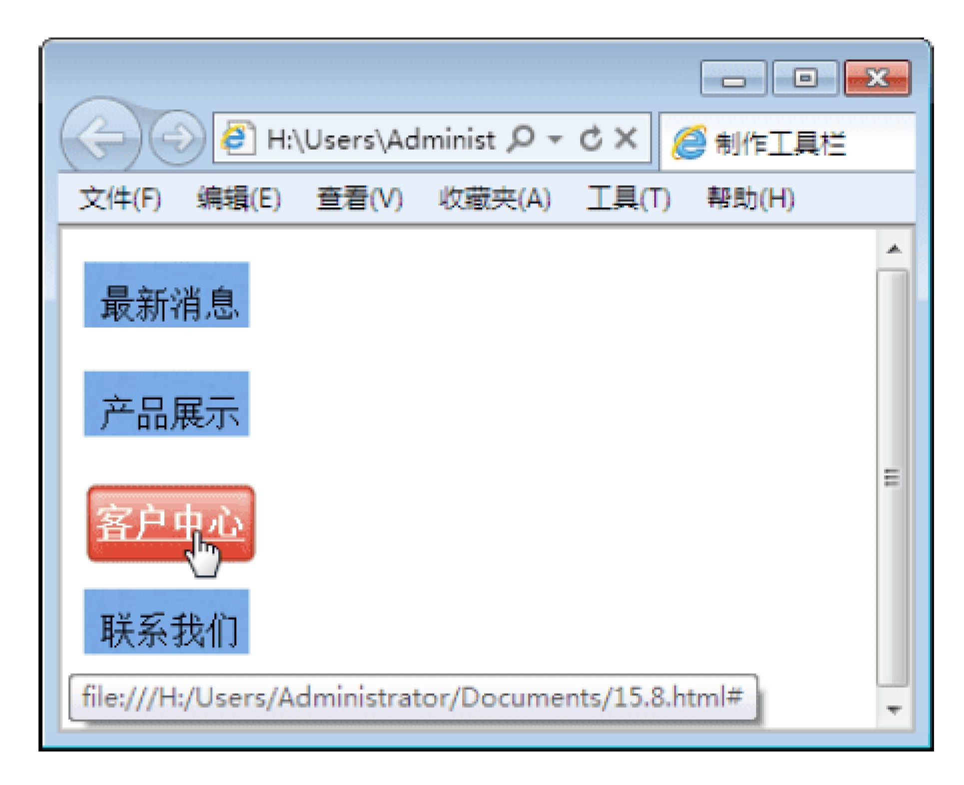


图 15-11　设置鼠标悬浮样式

15.4　综合实例 2——鼠标特效

在浏览网页时，我们看到鼠标指针的形状有箭头、手形和 I 字形，但在 Windows 环境下可以看到的鼠标指针样式要比这个多得多。CSS 弥补了 HTML 语言在这方面的不足，可以通

过 cursor 属性设置各种样式的鼠标，并且可以自定义鼠标。本实例将创建鼠标特效并自定义一个鼠标。

本实例结合前面介绍的内容，将创建一个鼠标特效实例。其具体步骤如下所示。

01 分析需求。

所谓鼠标特效，是指导背景图片、文字和鼠标指针发生变化，从而吸引人注意。本实例将创建 3 个超级链接，并设定它们的样式。实例完成后，在 IE 9.0 浏览器中的显示效果如图 15-12 和图 15-13 所示。

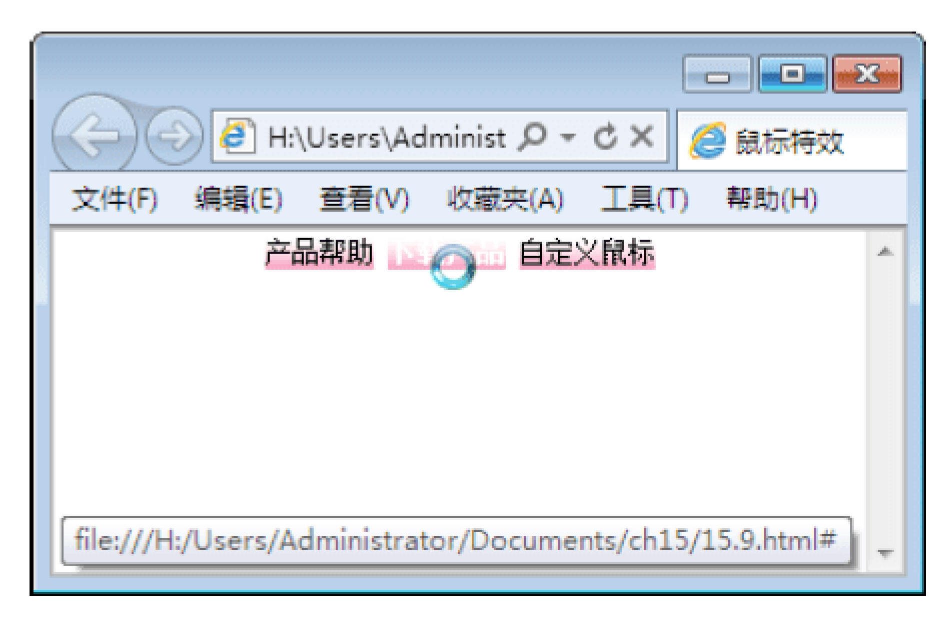


图 15-12　鼠标特效 1



图 15-13　鼠标特效 2

02 创建 HTML，实现基本超级链接。

```
<html >
<head>
<title>鼠标特效</title>
</head>
<body>
<center>
<a href="#" >产品帮助</a>
<a href="#" >下载产品</a>
<a href="#">自定义鼠标</a>
</center>
</body>
</html>
```

在 IE 9.0 中浏览效果如图 15-14 所示，可以在页面中看到 3 个超级链接，文字颜色为蓝色，并且带有下划线。

03 添加 CSS 代码，修饰整体样式。

```
<style type="text/css">
*{
    margin:0px;
    padding:0px;
}
```

```
body{
    font-family:"宋体";
    font-size:12px;
}
-->
</style>
```

在 IE 9.0 中浏览效果如图 15-15 所示，可以看到超级链接颜色不变，字号为 12 像素，字体为宋体。

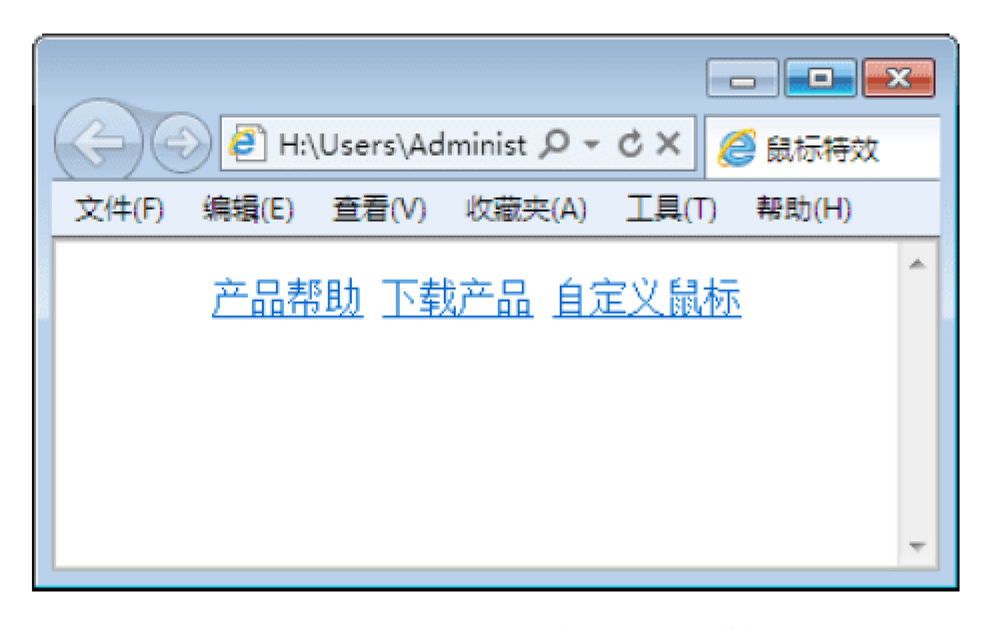


图 15-14　创建超级链接

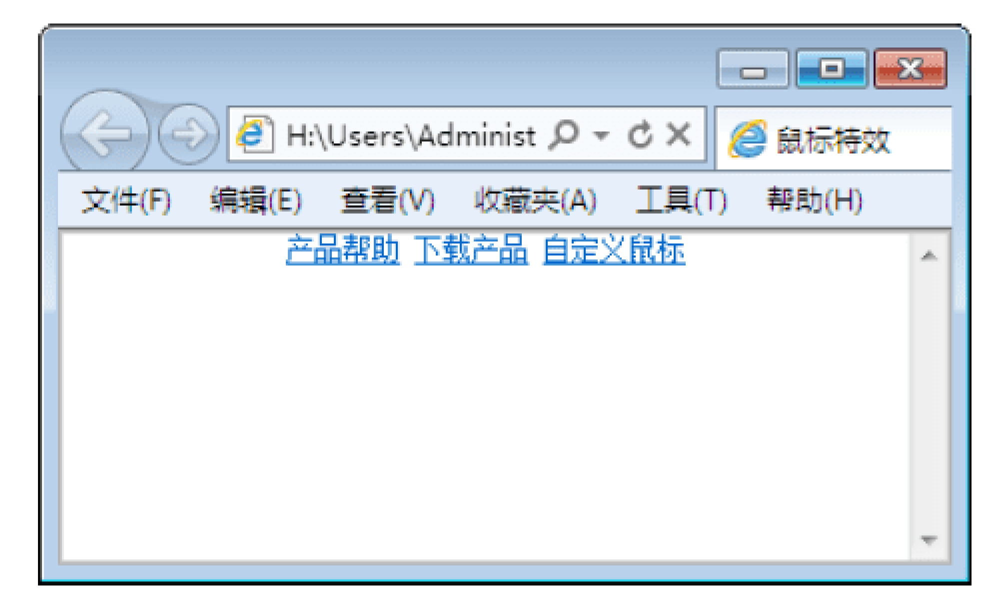


图 15-15　设置全局样式

04 添加 CSS 代码，修饰链接基本样式。

```
a, a:visited {
    line-height:20px;
     color: #000000;
     background-image:url(nav02.jpg);
     background-repeat: no-repeat;
     text-decoration: none;
}
```

在 IE 9.0 中浏览效果如图 15-16 所示，可以看到超级链接引入了背景图片，文字不带有下划线，并且颜色为黑色。

05 添加 CSS 代码，修饰悬浮样式。

```
a:hover {
     font-weight: bold;
     color: #FFFFFF;
}
```

在 IE 9.0 中浏览效果如图 15-17 所示，可以看到当鼠标放到超级链接上时，字体颜色变为白色，字体加粗。

06 添加 CSS 代码，设置鼠标指针。

```
<a href="#" style="cursor:help;">产品帮助</a>
<a href="#" style="cursor:wait;">下载产品</a>
```

```
<a href="#" style="CURSOR: url('0041.ani')">自定义鼠标</a>
```

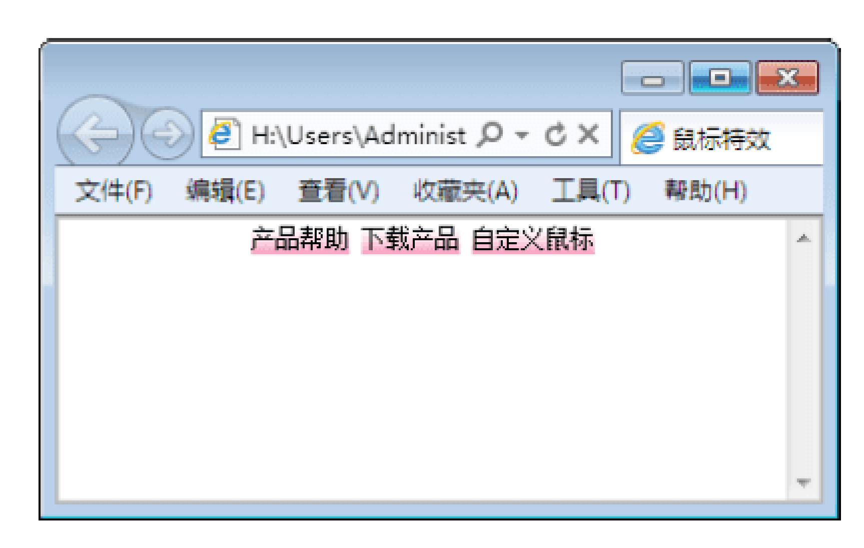



图 15-16　设置链接基本样式

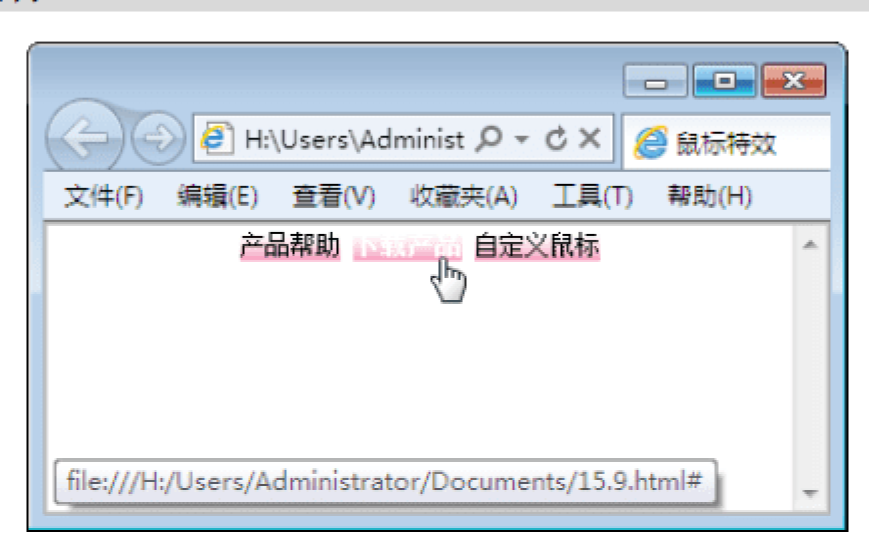


图 15-17　设置悬浮样式

在 IE 9.0 中浏览效果如图 15-18 所示，可以看到当鼠标放到超级链接上时，鼠标指针变为问号。

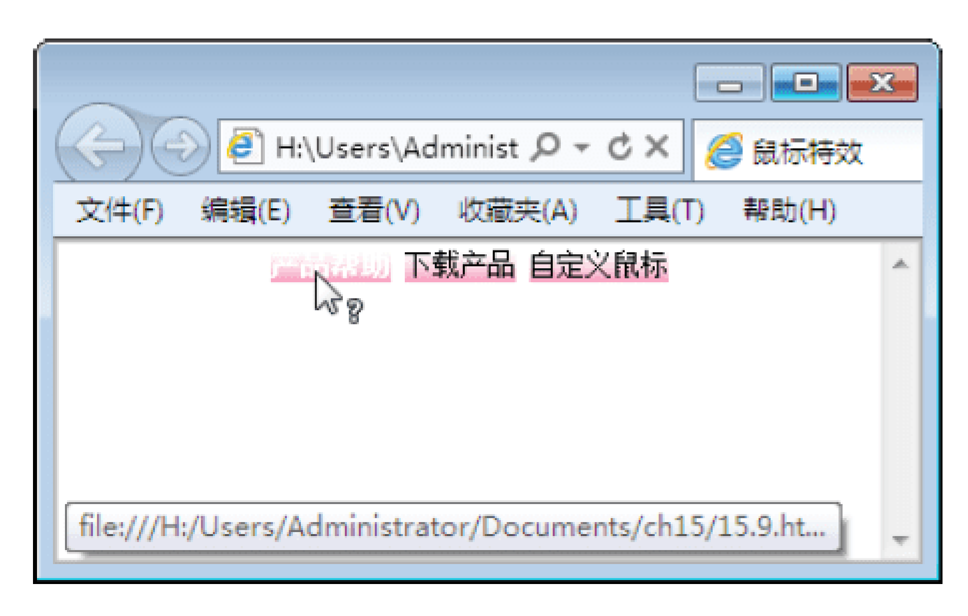


图 15-18　设置鼠标指针

15.5　专家解惑

1．丢失标记中的结尾斜线，会造成什么后果？

页面排版失效。结尾斜线也是造成页面失效比较常见的原因。我们很容易忽略结尾斜线之类的东西，特别是在 image 标签等元素中。在严格的 DOCTYPE 中这是无效的。要在 img 标签结尾处加上“/”以解决此问题。

2．设置了超级链接激活状态，怎么看不到结果？

当前激活状态“a:active”一般被显示的情况非常少，因此很少使用。因为当用户单击一个超级链接之后，焦点很容易从这个链接上转移到其他地方。例如，新打开的窗口等，此时该超级链接就不再是“当前激活”状态了。

3．为什么相同的鼠标效果在不同浏览器中不一样？

很多时候，浏览器调用的鼠标是操作系统的鼠标效果，因此同一用户浏览器之间的差别很小，但不同操作系统的用户之间还是存在差异的。例如有些鼠标效果可以在 IE 浏览器中显示，但不能在 Firefox 浏览器中显示。

第 16 章

用CSS美化表格和表单样式

HTML 数据表格和表单都是网页中常见的元素，表格通常用来显示二维关系数据和排版，从而达到页面整齐和美观的效果。而表单作为客户端和服务器交流的窗口，可以获取客户端信息，并反馈服务器端信息。

16.1　表格样式

在传统网页设计中，表格一直占据比较重要的地位，它用来对网页排版布局。基本上每个网页中都有一个表格来布局，它可以帮助设计者控制网页布局、控制内容的显示等。使用表格排版网页，可以使网页更美观，内容更清晰，更易于维护和更新。

16.1.1　表格边框样式

在显示一个表格数据时，通常都带有表格边框，用来界定不同单元格的数据。当表格的描述标记 border 值大于 0 时，则显示边框，如果 order 值为 0，则不显示边框。显示边框之后，可以使用 CSS 的 border 属性及衍生属性 border-collapse 对边框进行修饰，其中 border 属性表示对边框进行样式、颜色和宽度设置，从而达到提高样式效果的目的，这个属性前面已经介绍过了，其使用方法和前面一模一样，只不过修饰的对象变了。

Border-collapse 属性主要用来设置表格的边框是否被合并为一个单一的边框，还是像在标准 HTML 中那样分开显示。其语法格式为：

```
border-collapse : separate | collapse
```

其中 separate 是默认值，表示边框会被分开，不会忽略 border-spacing 和 empty-cells 属性。而 collapse 属性表示边框会合并为一个单一的边框，会忽略 border-spacing 和 empty-cells 属性。

【例 16.1】（实例文件：ch16\16.1.html）

```
<html>
<head>
<title>年度收入</title>
<style>
```

```
<!--
.tabelist{
        border:1px solid #429fff;     /*  表格边框  */
        font-family:"楷体";
        border-collapse:collapse;     /*  边框重叠  */
}
.tabelist caption{
        padding-top:3px;
        padding-bottom:2px;
        font-weight:bolder;
        font-size:15px;
        font-family:"幼圆";
        border:2px solid #429fff;     /*  表格标题边框  */
}
.tabelist th{
        font-weight:bold;
        text-align:center;
}
.tabelist td{
        border:1px solid #429fff;     /*  单元格边框  */
        text-align:right;
        padding:4px;
}
-->
</style>
</head>
<body>
<table class="tabelist">
        <caption class="tabelist">
        2011 季度  07-09
        </caption>
        <tr>
           <th>选项</th>
               <th>07 月</th>
               <th >08 月</th>
               <th>09 月</th>
        </tr>
        <tr>
               <td>收入</td>
               <td>8000</td>
               <td>9000</td>
               <td>7500</td>
```

```
        </tr>
        <tr>
            <td>吃饭</td>
            <td>600</td>
            <td>570</td>
            <td>650</td>
        </tr>
        <tr>
            <td>购物</td>
            <td>1000</td>
            <td>800</td>
            <td>900</td>
        </tr>
        <tr>
            <td>买衣服</td>
            <td>300</td>
            <td>500</td>
            <td>200</td>
        </tr>
        <tr>
            <td>看电影</td>
            <td>85</td>
            <td>100</td>
            <td>120</td>
        </tr>
        <tr>
            <td>买书</td>
            <td>120</td>
            <td>67</td>
            <td>90</td>
        </tr>
    </table>
    </body>
    </html>
```

在 IE 9.0 中浏览效果如图 16-1 所示，可以看到表格带有边框显示，其边框宽度为 1 像素，直线显示，并且有些边框进行了合并。表格标题“2011 季度 07-09”也带有边框显示，字号为 15 像素，字体为幼圆并加粗显示。表格中的每个单元格以 1 像素、直线的方式显示边框，并将显示对象右对齐。



图 16-1　表格样式修饰

对于上面的例子，我们会发现没有使用 HMTL 标记中的 border 设置边框，而是使用 CSS 的属性 border 来设置表格边框，这样做可以在不同的浏览器上显示相同样式。

16.1.2　表格边框宽度

虽然使用 HTML 标记的描述 border 也能增加表格的宽度，但我们还是推荐使用 CSS 属性来设置边框宽度。可使用 border-width 对边框宽度进行设置，从而更改显示样式。如果需要单独设置某一个边框宽度，可以使用 border-width 的衍生属性设置，如 border-top-width 和 border-left-width 等。

【例 16.2】（实例文件：ch16\16.2.tml）

```
<html>
<head>
<title>表格边框宽度</title>
<style>
table{
        text-align:center;
         width:500px;
         border-width:6px;
         border-style:double;
         color:blue;
}
td{
         border-width:3px;
         border-style:dashed;
}
</style>
</head>
```

```
<body>
<table border=1 cellspacing="3" cellpadding="0">
  <tr>
    <td>姓名</td>
    <td class=tds>性别</td>
    <td>年龄</td>
  </tr>
  <tr>
    <td>张三</td>
    <td>男</td>
    <td>31</td>
  </tr>
  <tr>
    <td>李四 </td>
    <td>男</td>
    <td>18</td>
  </tr>
</table>
</body>
</html>
```

在 IE 9.0 中浏览效果如图 16-2 所示，可以看到表格带有边框，宽度为 6 像素，显示样式为双线式，表格中字体颜色为蓝色。单元格边框宽度为 3 像素，显示样式为破折线式。



图 16-2 设置表格宽度

16.1.3 表格边框颜色

表格颜色设置非常简单，通常使用 CSS 属性 color 设置表格中的文本颜色，使用 background-color 设置表格背景色。如果为了突出表格中的某一个单元格，还可以使用 background-color 设置某一个单元格颜色。

【例 16.3】（实例文件：ch16\16.3.tml）

```
<html>
<head>
<title>表格边框色和背景色</title>
```

```
<style>
*{
      padding:0px;
      margin:0px;
}
body{
      font-family:"宋体";
      font-size:12px;
}
table{
      background-color:yellow;
      text-align:center;
      width:500px;
      border:1px solid green;
}
td{
      border:1px solid green;
       height:30px;
       line-height:30px;
}
.tds{
      background-color:blue;
}
</style>
</head>
<body>
<table   cellspacing="3" cellpadding="0">
   <tr>
      <td>姓名</td>
      <td class=tds>性别</td>
      <td>年龄</td>
   </tr>
   <tr>
      <td>张三</td>
      <td>男</td>
      <td>32</td>
   </tr>
   <tr>
      <td>莉莉  </td>
      <td>女</td>
      <td>28</td>
   </tr>
```

```
</table>
</body>
</html>
```

在 IE 9.0 中浏览效果如图 16-3 所示，可以看到表格带有边框，边框样式显示为绿色，表格背景色为黄色，其中一个单元格背景色为蓝色。

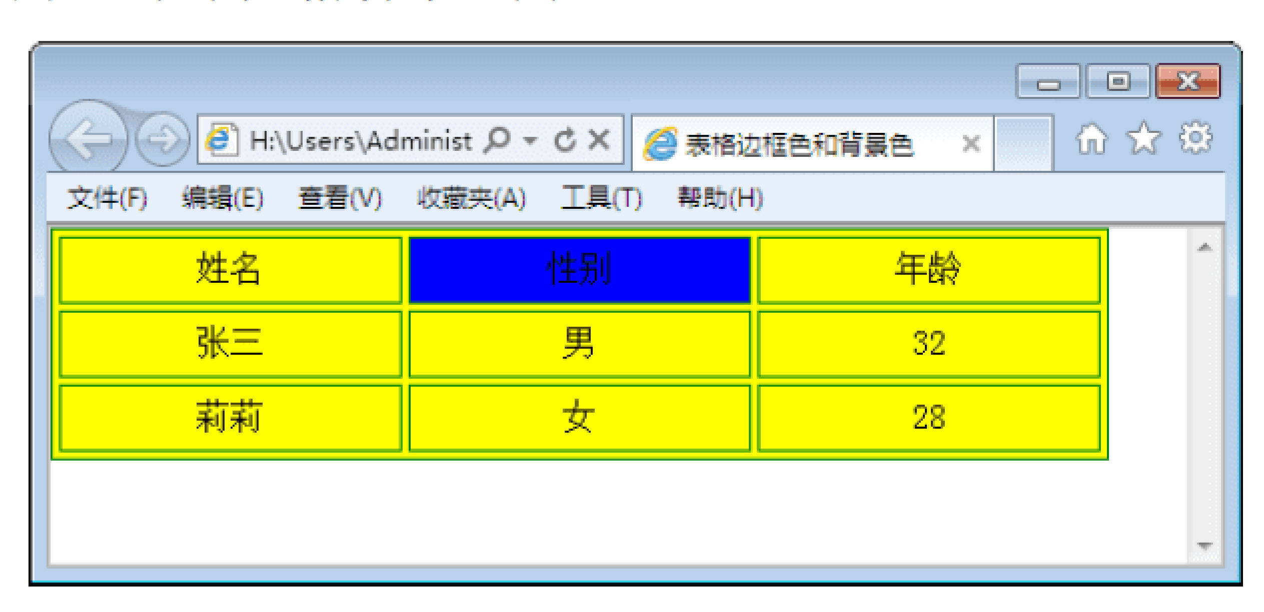


图 16-3　设置边框背景色

16.2　CSS 与表单

表单可以用来向 Web 服务器发送数据，尤其是经常用于主页页面——用户输入信息后将信息发送到服务器中。在 HTML 中实际使用的标记有 form、input、textarea、select 和 option。

16.2.1　美化表单中的元素

表单中元素非常多而且杂乱，如输入框、按钮、下拉菜单、单选按钮和复选框等。当使用表单将这些元素排列组合在一起的时候，其效果非常简单，这时设计者可以通过 CSS 相关样式，修饰表单输入框、文本框等元素外观。

在网页中，表单元素的背景色默认都是白色的，这样的背景色不能美化网页，所以可以使用颜色属性定义表单元素的背景色。表单元素背景色可以使用 background-color 属性定义，这样可以使表单元素不那么单调。使用示例如下所示。

```
input{
    background-color: #ADD8E6;
}
```

上面代码设置了 input 表单元素的背景色。

【例 16.4】（实例文件：ch16\16.4.tml）

```
<html>
<head>
<style>
<!--
input{                                        /*  所有 input 标记  */
    color: #cad9ea;
```

```
}
input.txt{                                /* 文本框单独设置 */
        border: 1px inset #cad9ea;
        background-color: #ADD8E6;
}
input.btn{                                /* 按钮单独设置 */
        color: #00008B;
        background-color: #ADD8E6;
        border: 1px outset #cad9ea;
        padding: 1px 2px 1px 2px;
}
select{
        width: 80px;
        color: #00008B;
        background-color: #ADD8E6;
        border: 1px solid #cad9ea;
}
textarea{
        width: 200px;
        height: 40px;
        color: #00008B;
        background-color: #ADD8E6;
        border: 1px inset #cad9ea;
}
-->
</style>
</head>
<body>
<h3>聊天室注册页面</h3>
<table border="1" width="45%">
<form method="post">
<tr><td width="30%">昵称:</td><td><input  class=txt>1－20 个字符<div id="qq"></div></td></tr>
<tr><td>密码:</td><td><input type="password" >长度为 6～16 位</td></tr>
<tr><td>确认密码:</td><td><input type="password" ></td></tr>
<tr><td>真实姓名：</td><td><input name="username1"></td></tr>
<tr><td>性别:</td><td><select><option>男</option><option>女</option></select></td></tr>
<tr><td>E-mail 地址:</td><td><input value="sohu@sohu.com"></td></tr>
<tr><td>备注:</td><td><textarea cols=35 rows=10></textarea></td></tr>
<tr><td><input type="button" value="提交" class=btn /></td><td><input type="reset" value="重填"/>
</td></tr>
</form>
</table>
```

```
</body>
</html>
```

在 IE 9.0 中浏览效果如图 16-4 所示，可以看到表单中【昵称】输入框、【性别】下拉框和【备注】文本框中都显示了指定的背景颜色。

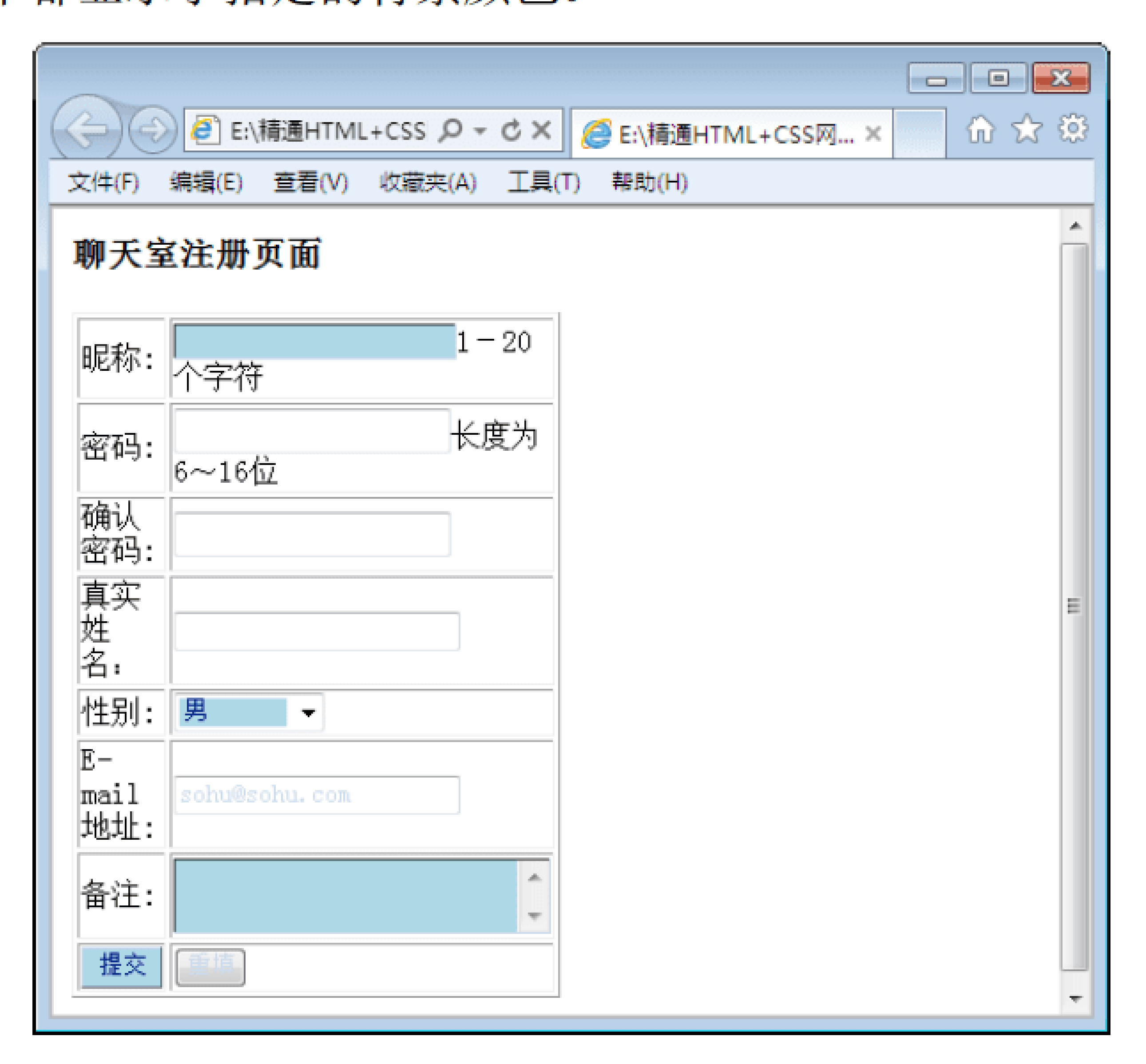


图 16-4　美化表单元素

在上面的代码中，首先使用 input 标记选择符定义了 input 表单元素的字体输入颜色，下面分别定义了两个类 txt 和 btn，txt 用来修饰输入框样式，btn 用来修饰按钮样式。最好分别定义 select 和 textarean 的样式，其样式定义主要涉及边框和背景色。

16.2.2　美化提交按钮

在网页设计中，还可以使用 CSS 属性来定义表单元素的边框样式，从而改变表单元素的显示效果。例如，可以将输入框的上、左和右边框去掉，形成和签名效果一样的输入框，或者将按钮的 4 个边框去掉，只剩下和超级链接一样的按钮。

对表单元素边框定义，可以采用 border-style、border-width 和 border-color 及其衍生属性。如果要对表单元素背景色进行设置，可以使用 background-color 属性，其中将值设置为 transparent（透明色）是最常见的一种方式。使用示例如下所示。

```
background-color:transparent;        /* 背景色透明 */
```

【例 16.5】（实例文件：ch16\16.5tml）

```
<html>
<head>
<title>表单元素边框设置</title>
<style>
```

```
<!--
form{
    margin:0px;
    padding:0px;
    font-size:14px;
}
input{
    font-size:14px;
    font-family:"幼圆";
}
.t{
    border-bottom:1px solid #005aa7;        /* 下划线效果 */
    color:#005aa7;
    border-top:0px; border-left:0px;
    border-right:0px;
    background-color:transparent;           /* 背景色透明 */
}
.n{
    background-color:transparent;           /* 背景色透明 */
    border:0px;                             /* 边框取消 */
}
-->
</style>
  </head>
<body>
<center>
<h1>签名页</h1>
<form method="post">
    值班主任: <input  id="name" class="t">
    <input type="submit" value="提交上一级签名>>" class="n">
</form>
</center>
</body>
</html>
```

在 IE 9.0 中浏览效果如图 16-5 所示，可以看到输入框只剩下下边框，其他边框被去掉了，提交按钮只剩下了显示文字，常见的矩形形式被去掉了。

在上面的代码中，样式表中定义了两个类标识符 t 和 n，t 用来设置输入框显示样式，此处设置输入框的左、上、下 3 个边框宽度为 0，并设置了输入框输入字体颜色为浅蓝色，下边框宽度为 1 像素，直线样式显示，颜色为浅蓝色。在类标识符 n 中，设置背景色为透明色和边框宽度为 0，这样就去掉了按钮常见的矩形样式。

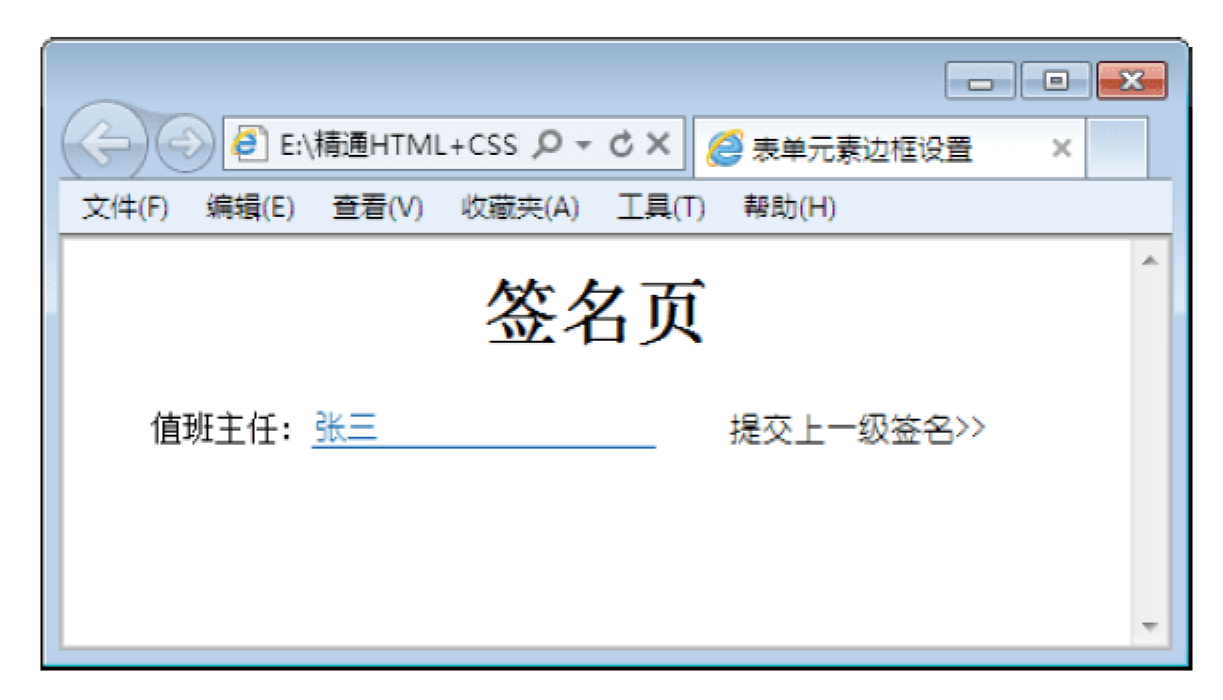


图 16-5　表单元素边框设置

16.2.3　美化下拉菜单

在网页设计中，有时为了突出效果，会对文字进行加粗等设置，以便用户注意到这些重要文字。同样也可以对表单元素中的文字进行修饰，从而达到突出的效果。下拉菜单是表单元素中常用的元素之一，其样式设置非常重要。

CSS 属性不仅可以控制下拉菜单的整体字体和边框，还可以对下拉菜单中的每一个选项设置背景色和字体颜色。可以使用 font 相关属性对字体进行设置，例如 font-size、font-weight 等，可以采用 color 和 background-color 属性对颜色进行设置。

【例 16.6】（实例文件：ch16\16.6.tml）

```
<html>
<head>
<title>美化下拉菜单</title>
<style>
<!--
.blue{
      background-color:#7598FB;
      color: #000000;
      font-size:15px;
      font-weight:bolder;
      font-family:"幼圆";
}
.red{
      background-color:#E20A0A;
      color: #ffffff;
      font-size:15px;
      font-weight:bolder;
      font-family:"幼圆";
}
.yellow{
       background-color:#FFFF6F;
```

```
        color: #000000;
        font-size:15px;
        font-weight:bolder;
        font-family:"幼圆";
    }
    .orange{
        background-color:orange;
        color:#000000;
        font-size:15px;
        font-weight:bolder;
        font-family:"幼圆";
    }
    -->
    </style>
      </head>
    <body>
    <form method="post">
        <p><label for="color">选择暴雨预警信号级别:</label>
        <select name="color" id="color">
            <option value="">请选择</option>
            <option value="blue" class="blue">暴雨蓝色预警信号</option>
            <option value="yellow" class="yellow">暴雨黄色预警信号</option>
            <option value="orange" class="orange">暴雨橙色预警信号</option>
            <option value="red" class="red">暴雨红色预警信号</option>
        </select></p>
        <p><input type="submit" value="提交"></p>
    </form>
    </body>
    </html>
```

在 IE 9.0 中浏览效果如图 16-6 所示，可以看到下拉菜单中的每个菜单项显示不同的背景色，用以和其他菜单项区别。使用这种方式显示选项，会提高人的注意力，减少犯错的机会。

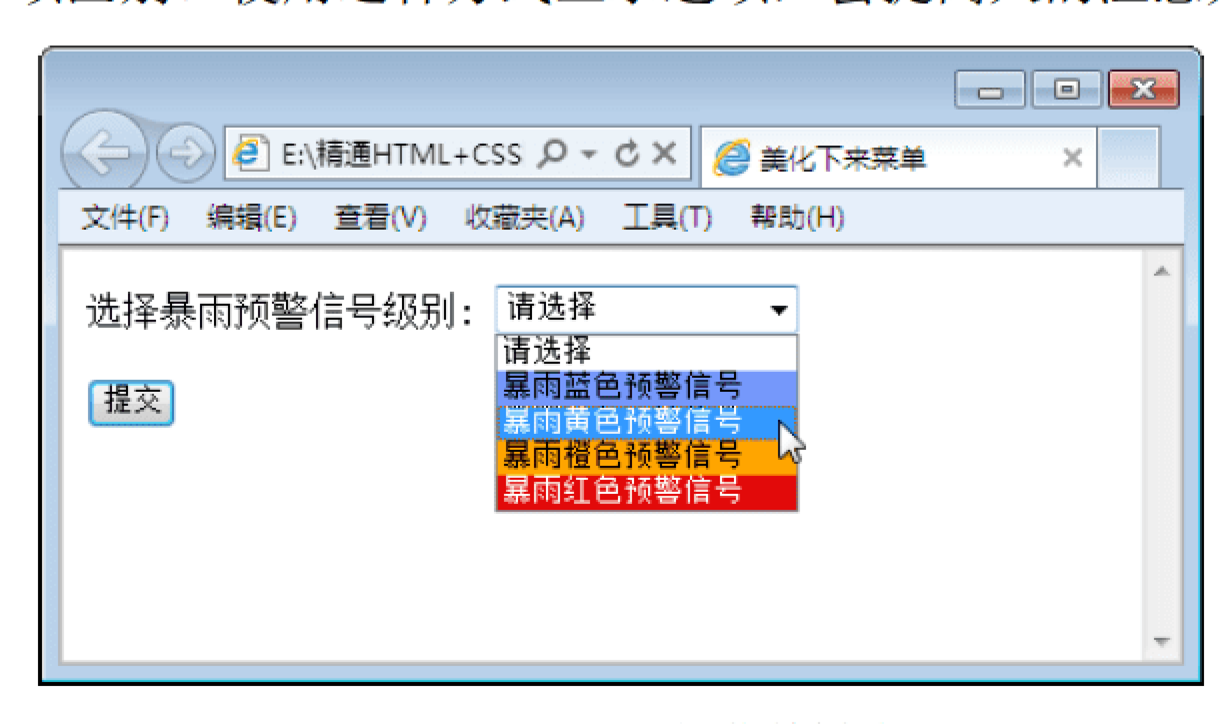


图 16-6　设置下拉菜单样式

在上面的代码中，设置了 4 个类标识符，用来对应不同的菜单选项。其中每个类中都设置了选项的背景色、字体颜色、大小和字形。

16.3 综合实例 1——隔行变色

当使用表格显示大量数据的时候，由于表格的行和列比较多，若此时单元格采用相同的背景色，用户在查看数据时会感到非常凌乱，并且读数据时容易出错。通常的解决方法就是采用隔行变色，使得奇数行和偶数行的背景色不一样，从而使数据一目了然。本节将结合前面学习的知识，创建一个隔行变色实例。具体操作步骤如下。

01 分析需求。

如果要实现表格隔行变色，首先需要实现一个表格，定义其显示样式，然后再设置其奇数行和偶然行显示的颜色即可。实例实现后，其显示效果如图 16-7 所示。

02 创建 HTML 页面，实现基本 table 表格。

```
<html>
<head>
<title>隔行变色</title>
</head>
<body>
<h1>设计优雅数据表格</h1>
<table border=1>
<tr>
<th>排名</th>
<th>姓名</th>
<th>总分</th>
<th>语文</th>
<th>数学</th>
</tr>
<tr><td>1</td><td>孔　宇</td><td>180</td><td>91</td><td>89</td></tr>
<tr><td>2</td><td>曹圆新</td><td>176</td><td>76</td><td>100</td></tr>
<tr><td>3</td><td>史雅琪</td><td>168</td><td>83</td><td>85</td></tr>
<tr><td>4</td><td>曹秀英</td><td>153</td><td>73</td><td>80</td></tr>
<tr><td>5</td><td>杨　青</td><td>146</td><td>70</td><td>76</td></tr>
</table>
</body>
</html>
```

在 IE 9.0 中浏览效果如图 16-8 所示，可以看到页面中显示了一个表格，其字体、边框等都是默认设置。



图 16-7　隔行变色效果

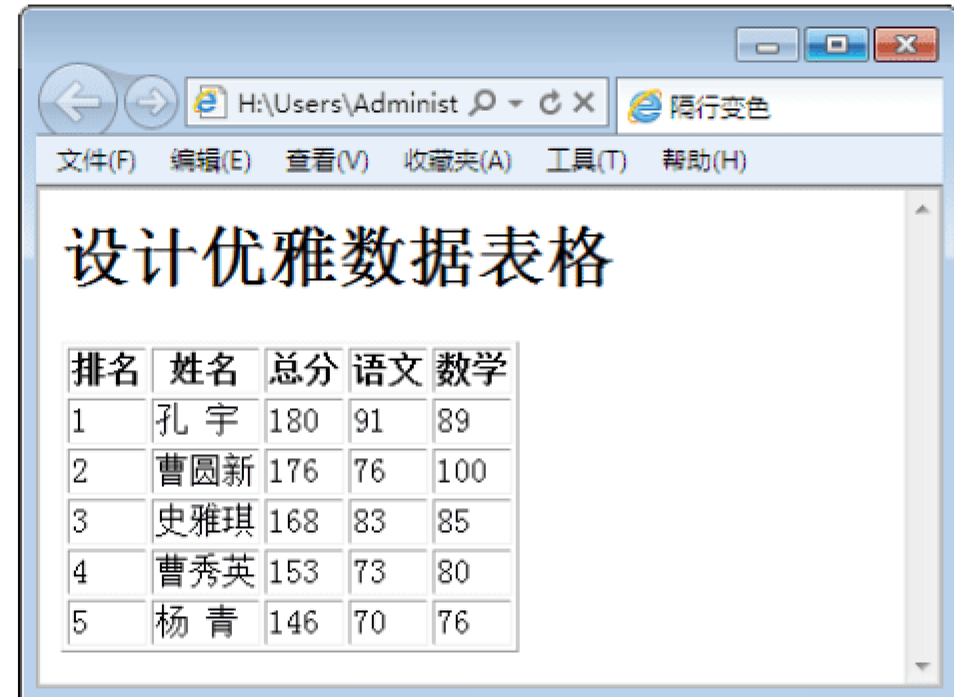


图 16-8　设置表格

03 添加 CSS 代码，设置标题和表格基本样式。

```
<style>
h1{font-size:16px;}
table{
    width:100%;
    font-size:12px;
    table-layout:fixed;
    empty-cells:show;
    border-collapse:collapse;
    margin:0 auto;
    border:1px solid #cad9ea;
    color:#666;
}
</style>
```

代码中设置了标题字体大小为 16 像素，表格字体大小为 12 像素，边框合并，边框大小为 1 像素，直线显示。其中 empty-cells 属性设置是否显示表格中的空单元格（仅用于“分离边框”模式），show 表示显示，hidden 表示隐藏。table-layout:fixed 语句表示表格的水平布局仅仅基于表格的宽度、表格边框的宽度、单元格间距、列的宽度，而和表格内容无关。

在 IE 9.0 中浏览效果如图 16-9 所示，可以看到页面中显示了一个表格，其大小充满了整个页面，边框大小显示为浅蓝色。

04 添加 CSS 代码，修饰 td 和 th 单元格。

```
th{
    height:30px;
    overflow:hidden;
}
td{height:20px;}
td,th{
    border:1px solid #cad9ea;
```

```
        padding:0 1em 0;
    }
```

在 IE 9.0 中浏览效果如图 16-10 所示，可以看到表格中单元格高度加大，td 增加到 20 像素，th 增加到 30 像素。单元格还带有边框显示，大小为 1 像素，直线样式，颜色为浅蓝色。



图 16-9　设置表格和标题样式

图 16-10　设置单元格样式

05 添加 CSS 代码，实现隔行变色。

```
tr:nth-child(even){
        background-color:#f5fafe;
}
```

在这里使用结构伪类标识符，实现了表格的隔行变色。在 IE 9.0 中浏览效果如图 16-11 所示，可以看到表格中奇数行显示浅蓝色。

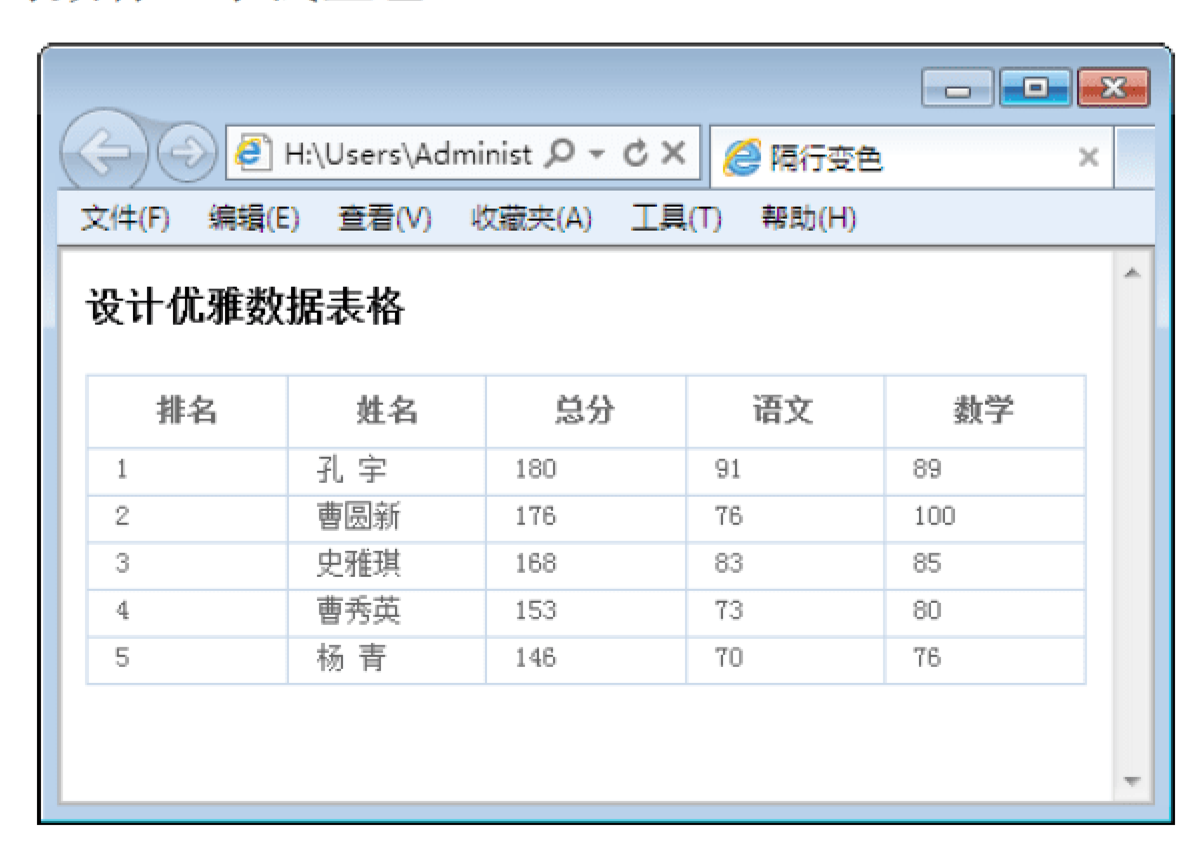


图 16-11　隔行变色

16.4　综合实例 2——表格图文网页布局

文字和图片等对象的定位是网页设计中的难点，用表格布局网页结构是一种非常重要，也是最为常用的方法，同时也是网页设计的重点。通过单元格可以对文本或图片进行定位，以达

到布局整齐的效果。本实例将结合前面学习的知识，实现通过表格对图片和文字进行布局。具体操作步骤如下。

01 分析需求。

使用表格进行排版，需要先确定图文竖排还是横排，这样就确定了图片的放置位置。如果采用竖排，可以创建一个表格，第一行放置图片，对应的下面放置文字介绍。最好使用 CSS 样式，对图片、文字和表格进行设定。实例完成后，效果如图 16-12 所示。

02 创建 HTML 页面，实现表格基本样式。

```
<html>
<head>
<title>无标题文档</title>
</head>
<body>
<table align=center border="1"   cellspacing="4" cellpadding="0">
  <tr>
    <td><img src="8.jpg" /><p>时尚大玩具！ 车展前实拍路虎极光 Coupe
</p>
</td>
    <td><img src="6.jpg" /><p>充满活力的小家伙  上海车展体验奥迪 Q3
</p></td>
  </tr>
  <tr>
    <td>路虎总是以硬汉的形象出现在人们面前</td>
    <td>奥迪将推出入门级 SUV 车型奥迪 Q3 时，大家都自然而然地觉得这款车型是和宝马 X1 竞争的
产品</td>
  </tr>
  <tr>
    <td> 全景天窗已经成为中高端 SUV 的流行趋势</td>
    <td>奥迪 Q3 相比两位兄长有简化也有复杂化 </td>
  </tr>
</table>
</body>
</html>
```

上面的代码创建了一个表格，第一行分别放置两个图片，下面两行放置文字。

在 IE 9.0 中浏览效果如图 16-13 所示，可以看到页面中显示了一个表格，表格中包含两张图片。



图 16-12　图文竖排

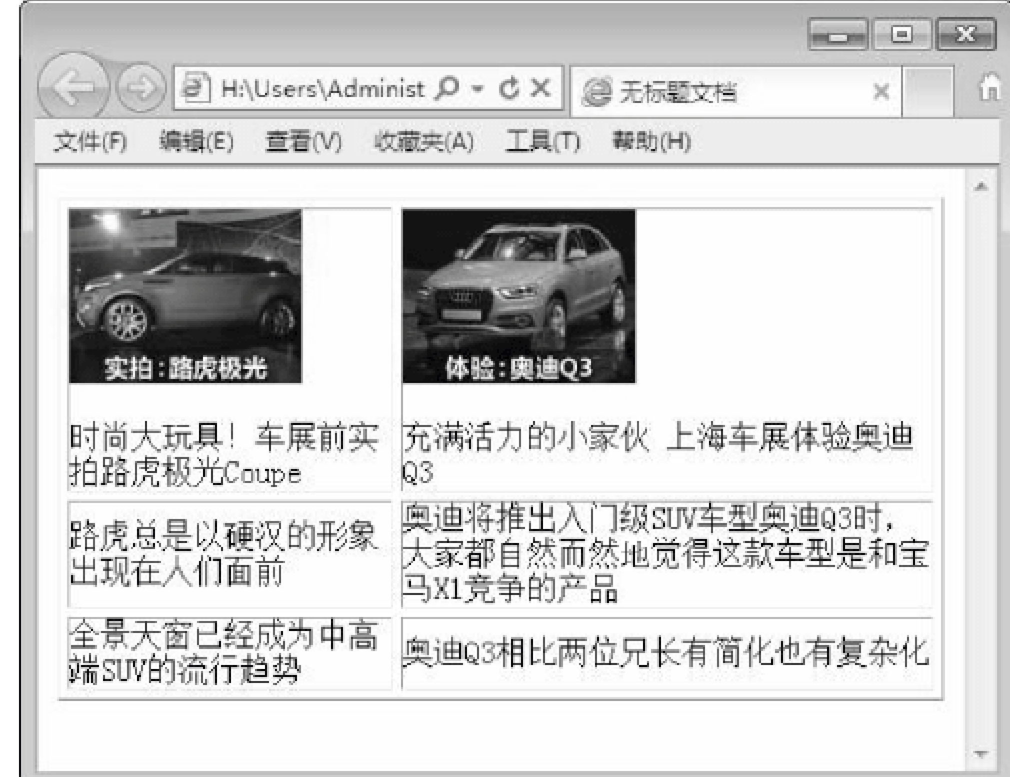


图 16-13　创建基本表格

03 添加 CSS 代码，修饰全局样式和表格。

```
<style>
* {
    padding:0px;
    margin:0px;
}
table {
    font-family:"幼圆";
    text-align:center;
    margin:10px 0 0 30px;
    background-color:#CCddCC;
    border-color:#0099ff;
    width:400px;
    font-size:15px;
}
</style>
```

上面的代码中，创建了一个 table 标记选择器，通过该选择器设置了字体样式和背景样式。

在 IE 9.0 中浏览效果如图 16-14 所示，可以看到表格的背景色为绿色，字体居中对齐，背景色为浅绿色，边框为浅蓝色。

04 添加 CSS 代码，修饰行 tr 和单元格 td。

```
tr {
    height:30px;
    line-height:30px;
}
.td{
    width:200px;
```

```
    }
```

上面代码定义了表格行的高度和文字行高，以及单元格宽度。

在 IE 9.0 中浏览效果如图 16-15 所示，可以看到与图 16-14 相比，其表格高度增加，行高增加。

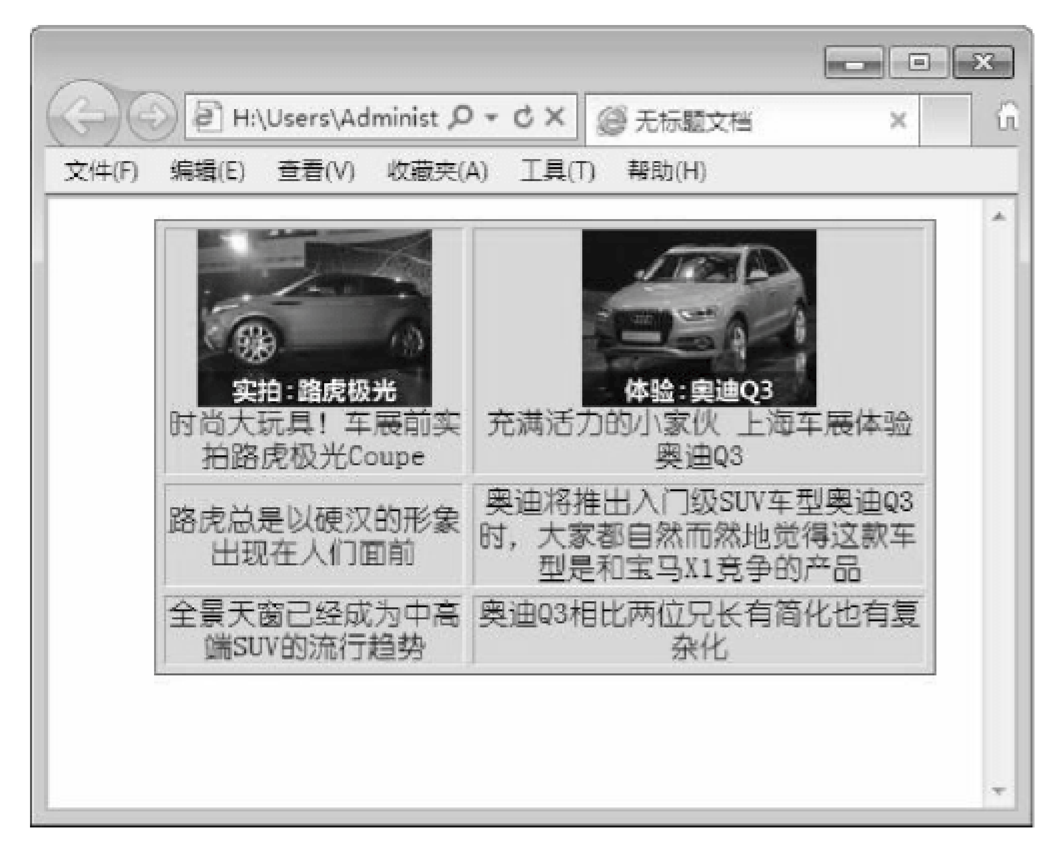


图 16-14　用 CSS 修饰表格



图 16-15　设置行高

16.5　综合实例 3——变色表格

如果长时间浏览大量数据和表格，即使使用了隔行变色，阅读时间长了仍然会让人感到疲劳。如果数据行能动态根据鼠标悬浮来改变颜色，就会使页面充满动态效果，不易让用户感到疲倦。

本实例将结合前面学习的知识，创建一个鼠标悬浮的变色表格。具体操作步骤如下。

01 分析需求。

首先需要建立一个表格，不单独设置行的背景色，统一采用表格本身的背景色。根据 CSS 设置可以实现该效果。仅仅通过 CSS 实现变色表格，只是对 Firefox 浏览器而言，如果是 IE 浏览器，需要结合使用 JavaScript。该实例完成后，效果如图 16-16 所示。

02 创建 HTML 网页，实现 table 表格。

```
<html>
<head>
<title>变色表格</title>
</head>
<body>
<table border="0" cellpadding="0" cellspacing="1">
<caption>
唐宋八大家
</caption>
```

```
    <tr>
      <th>姓名</th>
      <th>作品</th>
    </tr>
      <tr class="hui">
        <td>韩愈</td>
        <td>原道</td>
      </tr>
      <tr>
        <td>柳宗元</td>
        <td>三戒</td>
      </tr>
      <tr class="hui">
        <td>欧阳修</td>
        <td>醉翁亭记</td>
      </tr>
      <tr>
        <td>苏洵</td>
        <td>六国论</td>
      </tr>
      <tr class="hui">
        <td>苏轼</td>
        <td>水调歌头</td>
      </tr>
      <tr>
        <td>苏辙</td>
        <td>栾城集</td>
      </tr>
      <tr class="hui">
        <td>曾巩</td>
        <td>上欧阳舍人书</td>
      </tr>
      <tr>
        <td >王安石</td>
        <td>伤仲永</td>
      </tr>
  </table>
  </body>
  </html>
```

在 IE 9.0 中浏览效果如图 16-17 所示，可以看到页面中的表格不带边框，字体都是默认显示。

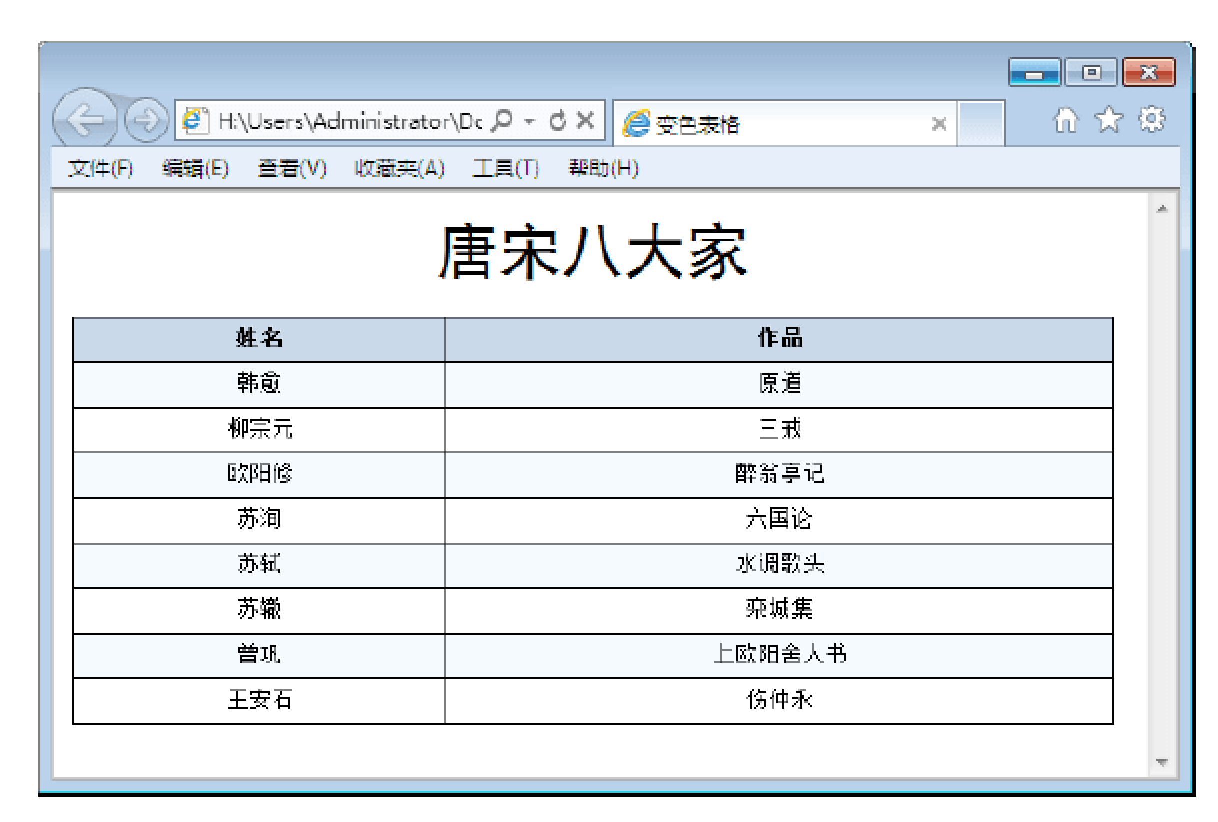


图 16-16　变色表格效果

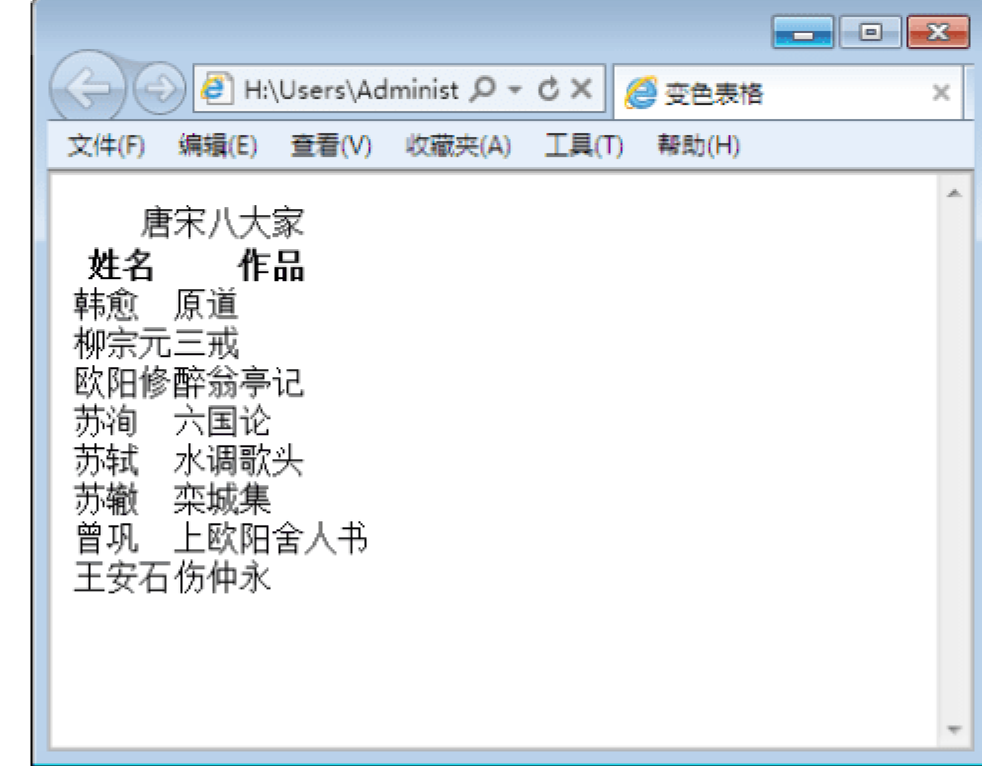


图 16-17　创建基本表格

03 添加 CSS 代码，修饰表格和单元格。

```
<style type="text/css">
<!--
table {
        width: 600px;
        margin-top: 0px;
        margin-right: auto;
        margin-bottom: 0px;
        margin-left: auto;
        text-align: center;
        background-color: #000000;
        font-size: 9pt;
}
td {
        padding: 5px;
        background-color: #FFFFFF;
}
-->
</style>
```

在 IE 9.0 中浏览效果如图 16-18 所示，可以看到页面中的表格带有边框，字体居中显示，但列标题背景色为黑色，其中的文字不能够显示。

04 添加 CSS 代码，修饰标题。

```
caption{
        font-size: 36px;
        font-family: "黑体", "宋体";
        padding-bottom: 15px;
}
```

```
tr{
    font-size: 13px;
    background-color: #cad9ea;
    color: #000000;
}
th{
    padding: 5px;
}
.hui td {
    background-color: #f5fafe;
}
```

上面的代码中，使用了类选择器 hui，来定义每个 td 行所显示的背景色，此时需要在表格中每个奇数行都引入该类选择器，例如<tr class="hui">，从而设置奇数行背景色。这和综合实例 1 中对奇数行背景色的设置方式是不一样的。

在 IE 9.0 中浏览效果如图 16-19 所示，可以看到表格中列标题一行背景色显示为浅蓝色，并且表格中奇数行背景色为浅灰色，而偶数行背景色为默认的白色。

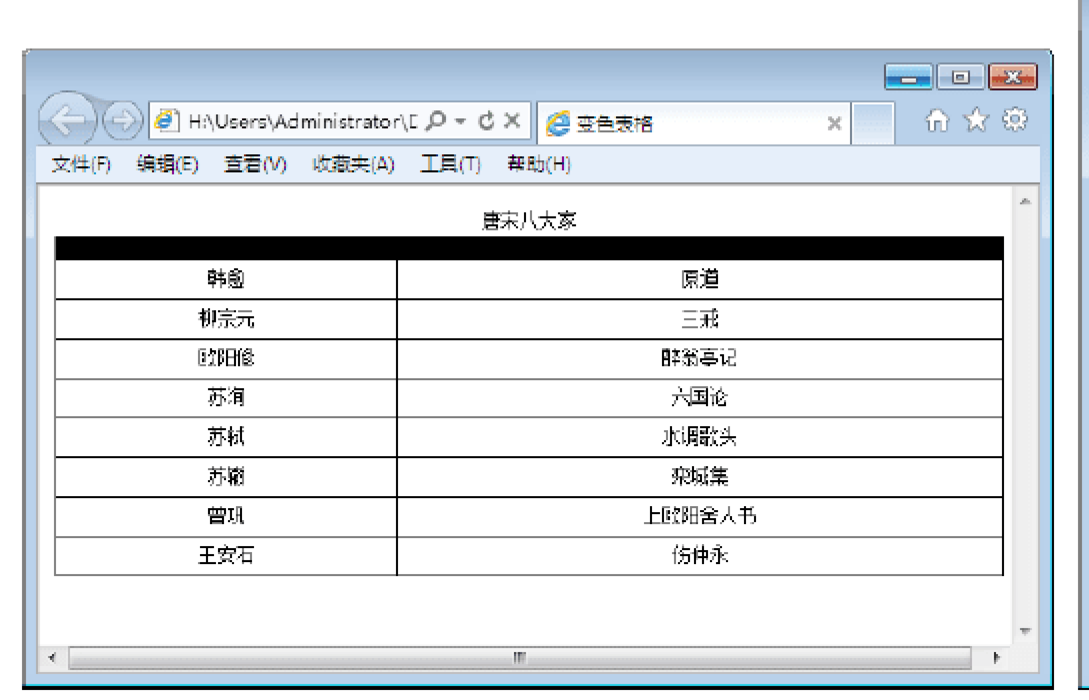


图 16-18　设置样式

图 16-19　设置奇数行背景色

16.6　综合实例 4——登录表单

在设计网页时，构建一个登录表单是非常常见的事情。登录表单在验证用户身份时扮演着重要的角色，可以保护一些敏感数据不被非法用户访问。设计和构建登录表单时，要尽量使其简单，并且要避免将一个表单用于多个任务。

本实例将结合前面学习的知识，创建一个简单的登录表单，具体操作步骤如下。

01 分析需求。

创建一个登寻表单，需要包含 3 个表单元素，一个名称输入框，一个密码输入框和两个按钮。然后添加一些 CSS 代码，对表单元素进行修饰即可。实例完成后，其效果如图 16-20 所示。

02 创建 HTML 网页，实现表单。

```
<html>
<head>
<title>用户登录</title>
<body>
<div>
<h1>用户登录</h1>
 <form action="" method="post">
姓名：<input type="text" id=name   />
密码：<input type="password" id=password name="ps"   />
<input type=submit value="提交" class=button>
<input type=reset value="重置" class=button>
</form>
</div>
</body>
</html>
```

在上面的代码中，创建了一个 div 层用来包含表单及其元素。

在 IE 9.0 中浏览效果如图 16-21 所示，可以看到显示了一个表单，其中包含两个输入框和两个按钮，输入框用来获取姓名和密码，按钮分别为提交按钮和重置按钮。

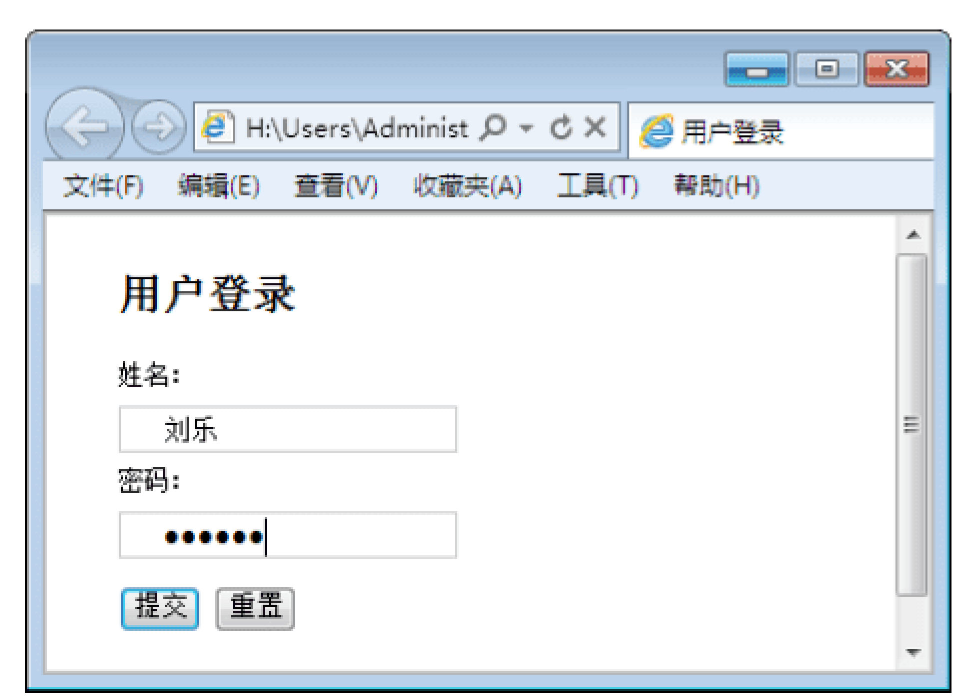


图 16-20　登录表单



图 16-21　创建登录表单

03 添加 CSS 代码，修饰标题和层。

```
<style>
h1{
      font-size:20px;
}
div{
      width:200px;
      padding:1em 2em 0 2em;
      font-size:12px;
```

```
}
</style>
```

上面代码中，设置了标题大小为 20 像素，div 层宽度为 200 像素，层中字体大小为 12 像素。在 IE 9.0 中浏览效果如图 16-22 所示，可以看到标题变小，并且密码输入框换行显示，其布局比原来的图片更加美观合理。

04 添加 CSS 代码，修饰输入框和按钮。

```
#name,#password{
    border:1px solid #ccc;
    width:160px;
    height:22px;
    padding-left:20px;
    margin:6px 0;
    line-height:20px;
}
.button{margin:6px 0;}
```

在 IE 9.0 中浏览效果如图 16-23 所示，可以看到输入框长度变短，输入框边框变小，并且表单元素之间距离增大，页面布局更加合理。

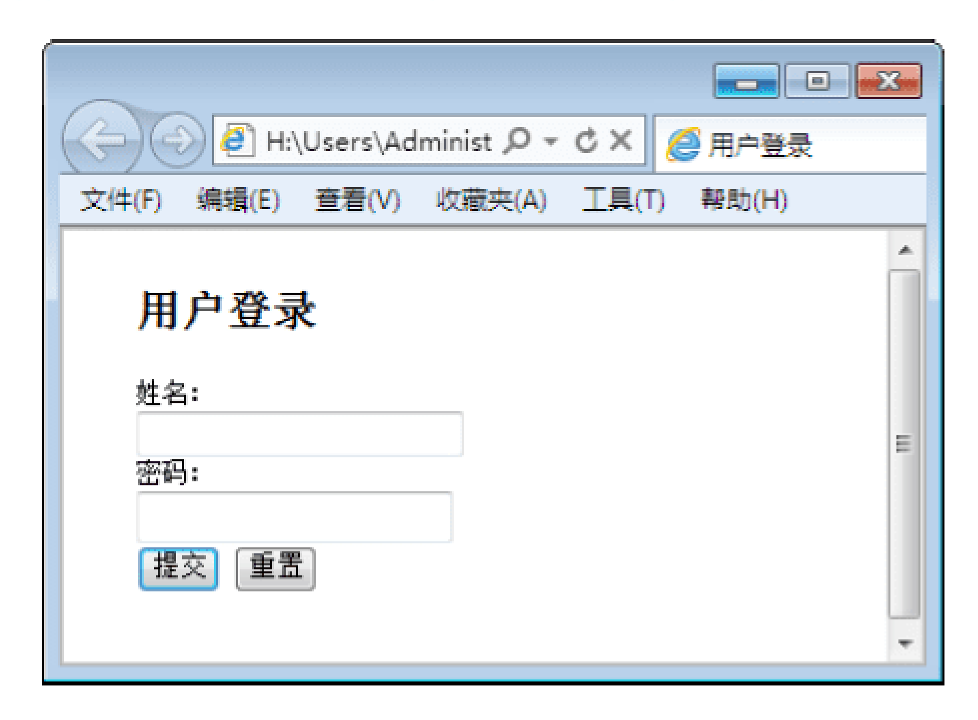


图 16-22　设置层大小

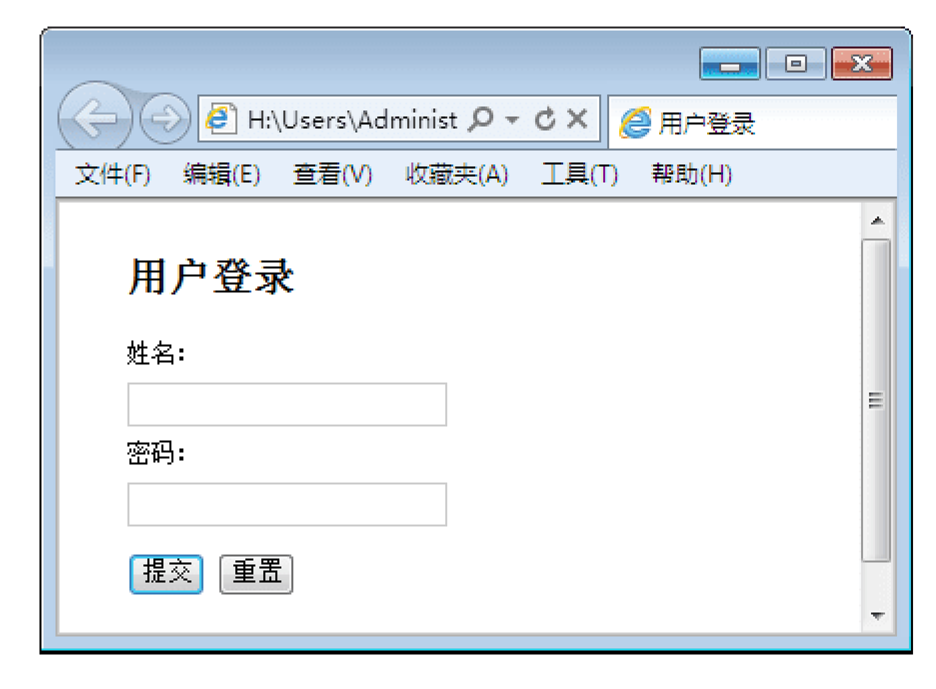


图 16-23　用 CSS 修饰输入框

16.7　综合实例 5——注册表单

不管是在线交易验证、评论新文章，还是管理某个应用，Web 表单总会出现在人们的视线中。在网页上，Web 表单把用户、信息、Web 产品或者服务连接了起来。它们能促进销售、捕捉用户行为、建立沟通与交流，以及更多。

本实例中，将使用一个表单内的各种元素来开发网站的注册页面，并用 CSS 样式来美化页面效果。具体操作步骤如下。

01 分析需求。

注册表单非常简单，通常包含三个部分，需要在页面上方给出标题，标题下方是正文部分，即表单元素，最下方是表单元素提交按钮。在设计这个页面时，需要把“用户注册”标题设置成 H1 大小，正文使用 p 来限制表单元素。实例完成后，实际效果如图 16-24 所示。

02 构建 HTML 页面，实现基本表单。

```
<html>
<head>
<title>注册页面</title>
</head>
<body>
<h1 align=center>用户注册</h1>
<form method="post" >
<p>姓    名：
<input type="text" class=txt size="12" maxlength="20" name="username" />
</p><p>性    别：
<input type="radio" value="male" />男
<input type="radio" value="female" />女
</p><p>年    龄：
<input type="text" class=txt name="age"   />
</p>
<p>联系电话：
<input type="text" class=txt name="tel" />
</p><p>电子邮件：
<input type="text" class=txt name="email" />
</p><p>联系地址：
<input type="text"   class=txt name="address" />
</p>
<p>
<input type="submit" name="submit" value="提交" class=but />
<input type="reset" name="reset" value="清除" class=but   />
</p>
</form>
</body>
</html>
```

在 IE 9.0 中浏览效果如图 16-25 所示，可以看到页面中创建了一个注册表单，包含标题“用户注册”以及“姓名”、“性别”、“年龄”、“联系方式”、“电子邮件”、“地址”等输入框和“提交”按钮等。其显示样式为默认样式。

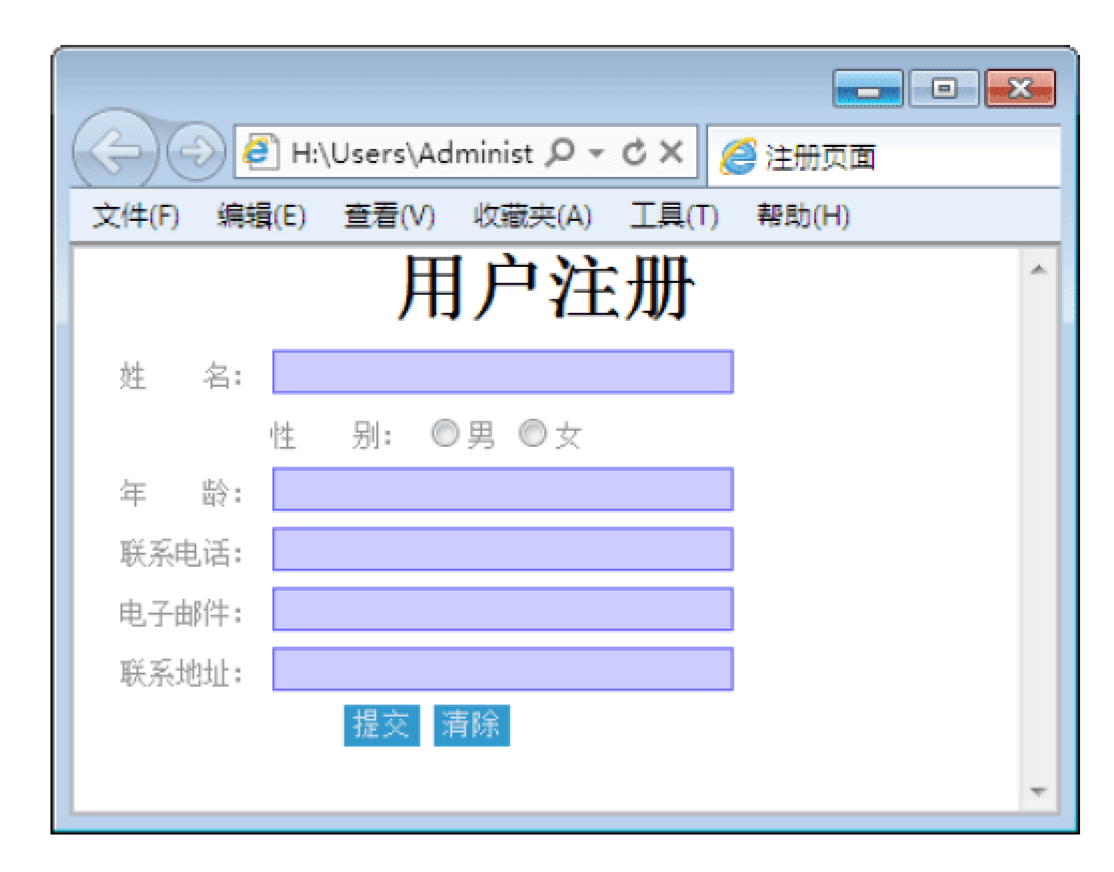


图 16-24 注册页面

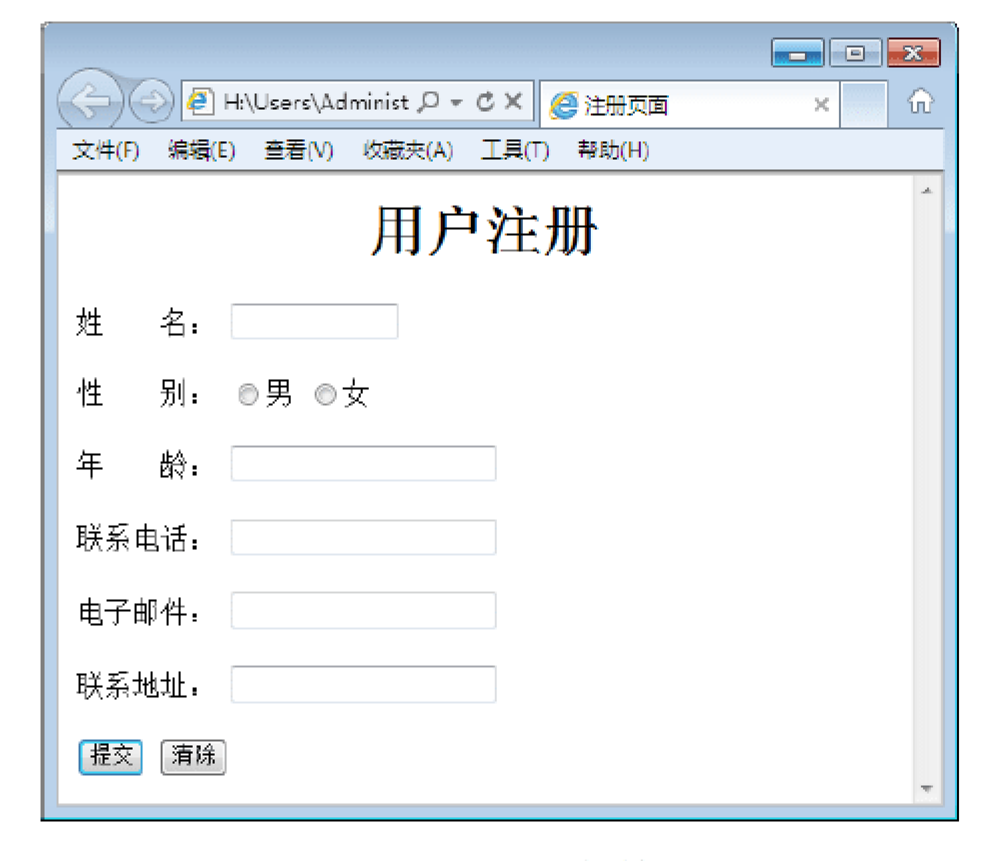


图 16-25 注册表单显示

03 添加 CSS 代码，修饰全局样式和表单样式。

```
<style>
*{
      padding:0px;
      margin:0px;
      }
body{
      font-family:"宋体";
      font-size:12px;
      }
form{
      width:300px;
      margin:0 auto 0 auto;
      font-size:12px;
      color:#999;
}
</style>
```

在 IE 9.0 中浏览效果如图 16-26 所示，可以看到与原来的页面相比，其表单元素之间距离变小。

04 添加 CSS 代码，修饰段落、输入框和按钮。

```
form p {
     margin:5px 0 0 5px;
     text-align:center;
}
.txt{
      width:200px;
      background-color:#CCCCFF;
      border:#6666FF 1px solid;
```

```
        color:#0066FF;
    }
    .but{
        border:0px#93bee2solid;
        border-bottom:#93bee21pxsolid;
        border-left:#93bee21pxsolid;
        border-right:#93bee21pxsolid;
        border-top:#93bee21pxsolid;*/
        background-color:#3399CC;
        cursor:hand;
        font-style:normal;
        color:#cad9ea;
    }
```

在 IE 9.0 中浏览效果如图 16-27 所示，可以看到表单元素带有背景色，其输入字体颜色为蓝色，边框颜色为浅蓝色。按钮带有边框，按钮上的文字颜色为浅色。

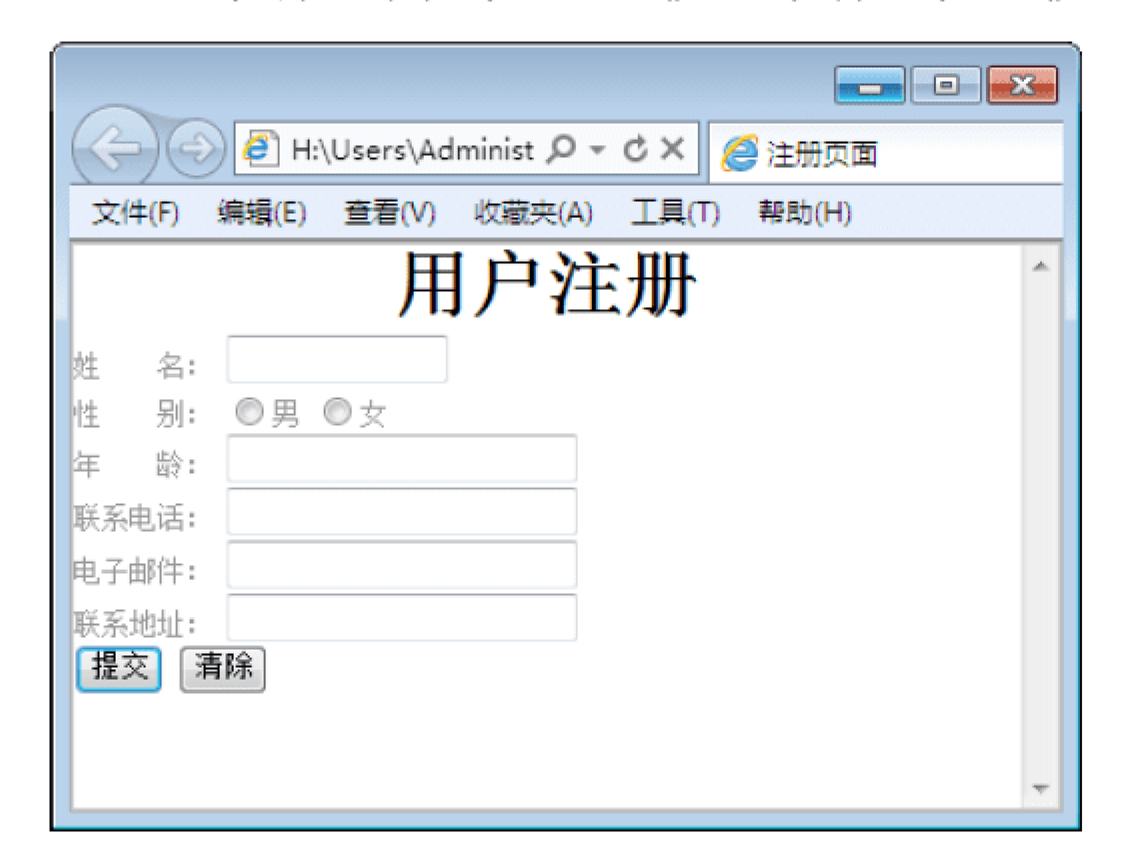


图 16-26　用 CSS 修饰表单样式

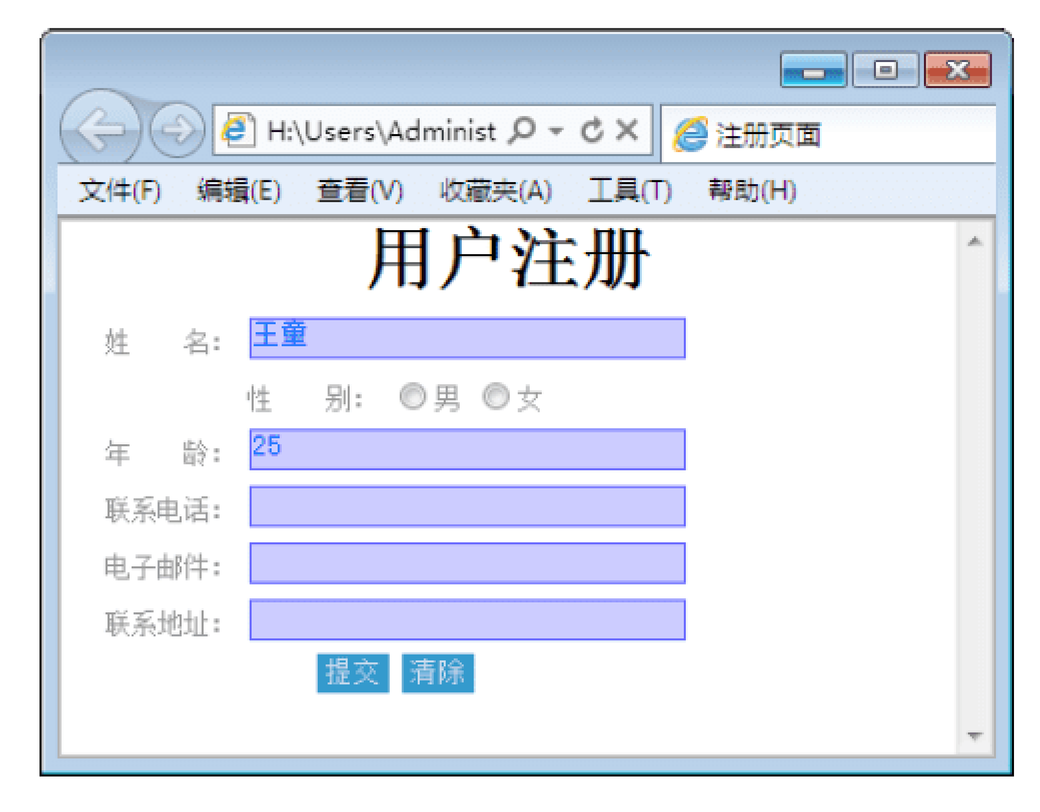


图 16-27　设置输入框和按钮样式

16.8　专家解惑

1. 构建一个表格时需要注意哪些方面？

在 HTML 页面中构建表格框架时，应该尽量遵循表格的标准标记，养成良好的编写习惯，并适当地利用 tab 键、空格和空行来提高代码的可读性，从而降低后期维护成本。特别是使用表格来布局一个较大的页面，需要在关键位置加上注释。

2. 使用 CSS 修饰表单元素时，采用默认值好还是使用 CSS 修饰好？

各个浏览器之间显示的差异，其中一个原因就是浏览器对部分 CSS 属性的默认值不同导致的，通常的解决办法就是指定该值，而不让浏览器使用默认值。

第 17 章

用CSS制作实用菜单

导航菜单是网站中必不可少的元素之一，通过导航菜单可以在页面上自由跳转。导航菜单风格往往影响网站整体风格，所以网页设计者会花费大量时间和精力制作各式各样的导航条，从而体现网站总体风格。利用 CSS 属性和项目列表，可以制作出美观大方的导航菜单。

17.1 项目列表

在 HTML 语言中，项目列表用来罗列显示一系列相关的文本信息，如类别信息，项目列表包括有序、无序和自定义列表。当引入 CSS 后，项目列表被赋予了很多新功能，并超越了最初的设计初衷，即采用项目列表和 CSS 制作漂亮的导航菜单。

17.1.1 美化无序列表

无序列表<ul>是网页中常见的元素之一，可使用<li>标记罗列各个项目，并且每个项目前面都带有特殊符号，例如黑色实心圆等。在 CSS 中，可以通过属性 list-style-type 来定义无序列表前面的项目符号。

对于无序列表，list-style-type 语法格式如下所示：

```
list-style-type : disc | circle | square | none
```

其中 list-style-type 参数值的含义，如表 17-1 所示。

表 17-1 list-style-type 属性的参数及其含义

参数	说明
disc	实心圆
circle	空心圆
square	实心方块
none	不使用任何标号

可以通过表 17-1 中的参数，为 list-style-type 设置不同的特殊符号，从而改变无序列表的样式。

【例 17.1】（实例文件：ch17\17.1.html）

```
<html>
<head>
<title>无序列表美化</title>
<style>
* {
      margin:0px;
      padding:0px;
      font-size:12px;
}
p {
       margin:5px 0 0 5px;
       color:#3333FF;
       font-size:14px;
       font-family:"幼圆";
}
div{
       width:300px;
       margin:10px 0 0 10px;
       border:1px #FF0000 dashed;
}
div ul {
       margin-left:40px;
       list-style-type: disc;
}
div li {
      margin:5px 0 5px 0;
      color:blue;
      text-decoration:underline;
}
</style>
</head>
<body>
<div class="big01">
   <p>焦点新闻</p>
   <ul>
      <li>适时补涂防晒霜  防晒护理才到位(1)11:36</li>
      <li>夏日凉爽爽  盘发才是王道(组图)10:39   </li>
      <li>浅色短裤+雪纺衫  清爽度过伏天儿 08:11   </li>
      <li>巧穿基础白衬衫  装扮高手必学法术 08:20 </li>
      <li>夏裙又有新花样  拼接图案最时髦 08:11 </li>
```



```
    </ul>
  </div>
  </body>
  </html>
```

在 IE 9.0 中的浏览效果如图 17-1 所示，可以看到页面中显示了一个导航栏，存在着不同的导航信息，每条导航信息前面都是使用实心圆作为每行信息的开头。在上面的代码中，使用 list-style-type 设置了无序列表中特殊符号为实心圆，border 设置了层 div 的边框显示为红色，虚线显示，宽度为 1 像素。

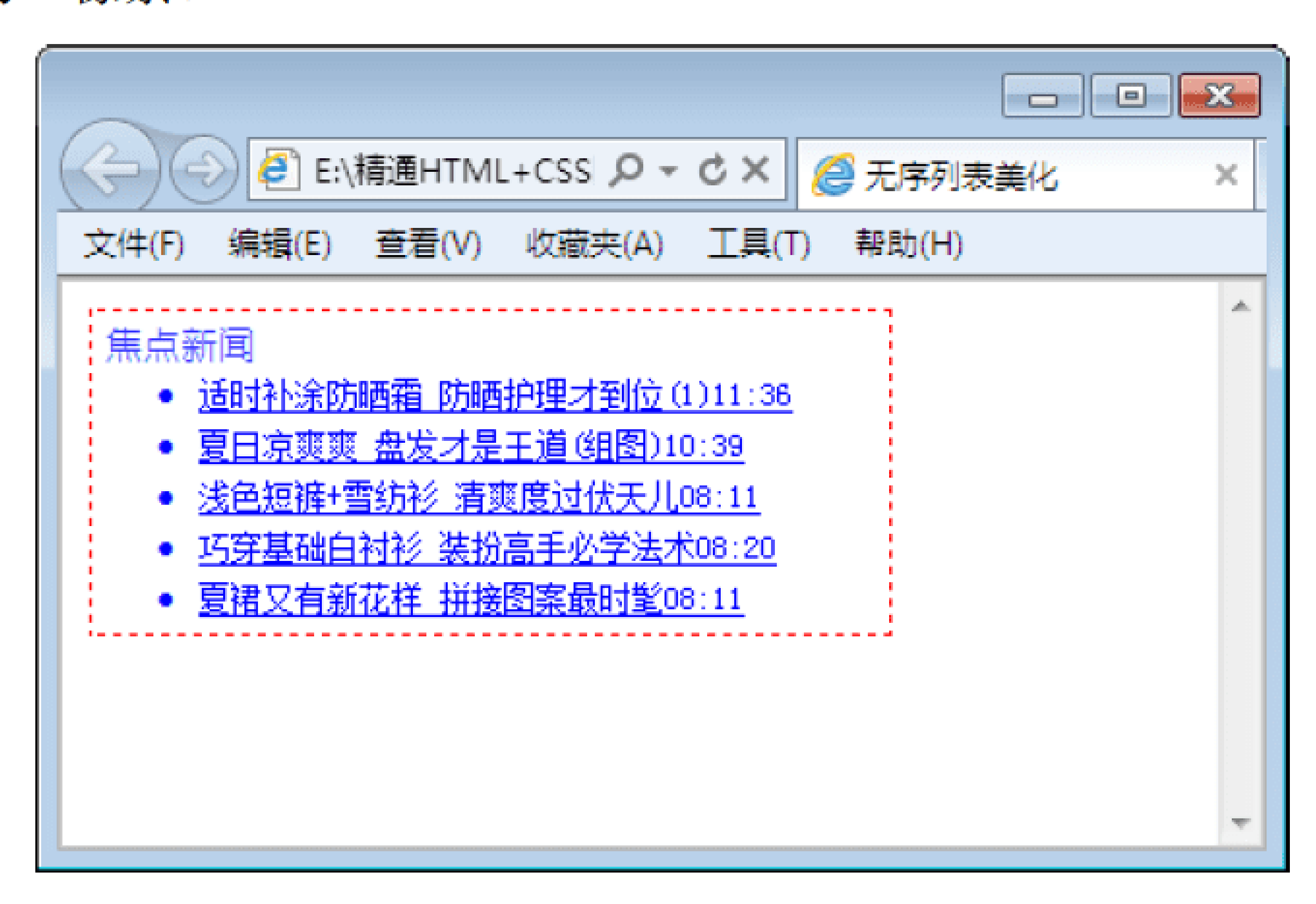


图 17-1　无序列表制作导航菜单

17.1.2　美化有序列表

有序列表标记<ol>可以创建具有顺序的列表，例如在每条信息前面加上 1、2、3、4 等。如果要改变有序列表前面的符号，同样需要使用 list-style-type 属性，只是属性值不同。

对于有序列表，list-style-type 语法格式如下所示：

```
list-style-type : decimal | lower-roman | upper-roman | lower-alpha | upper-alpha | none
```

其中 list-style-type 参数值的含义，如表 17-2 所示。

表 17-2　有序列表常用参数

参数	说明
decimal	阿拉伯数字
lower-roman	小写罗马数字
upper-roman	大写罗马数字
lower-alpha	小写英文字母
upper-alpha	大写英文字母
none	不使用项目符号

除了表 17-2 中的这些常用参数外，list-style-type 还具有很多不同的参数值。这里由于地域习惯，没有将一些罕见的项目符号罗列出来，例如传统的亚美尼亚数字、传统的乔治数字等。在对 list-style-type 的支持力度上，IE 9.0 做得很好。

【例 17.2】（实例文件：ch17\17.2.html）

```
<html>
<head>
<title>有序列表</title>
<style>
* {
        margin:0px;
        padding:0px;
        font-size:12px;
}
p {
       margin:5px 0 0 5px;
       color:#3333FF;
       font-size:14px;
       font-family:"幼圆";
       border-bottom-width:1px;
       border-bottom-style:solid;

}
div{
        width:300px;
        margin:10px 0 0 10px;
        border:1px #F9B1C9 solid;
}
div ol {
       margin-left:40px;
       list-style-type: decimal;
}
div li {
       margin:5px 0 5px 0;
       color:blue;
}
</style>
</head>
<body>
<div class="big">
   <p>热点追击</p>
   <ol>
```

```
        <li>损害 5 脏的坏习惯  水喝不够伤肾  </li>
        <li>  大暑时节应重视“以粥养生”    </li>
        <li>  巧穿基础白衬衫  装扮高手必学法术</li>
        <li>  夏裙又有新花样  拼接图案最时髦  </li>
    </ol>
</div>
</body>
</html>
```

在 IE 9.0 中的浏览效果如图 17-2 所示，可以看到页面中显示了一个导航栏，导航信息前面带有相应的数字，表示其顺序。导航栏具有红色边框，并用一条蓝线将题目和内容分开。

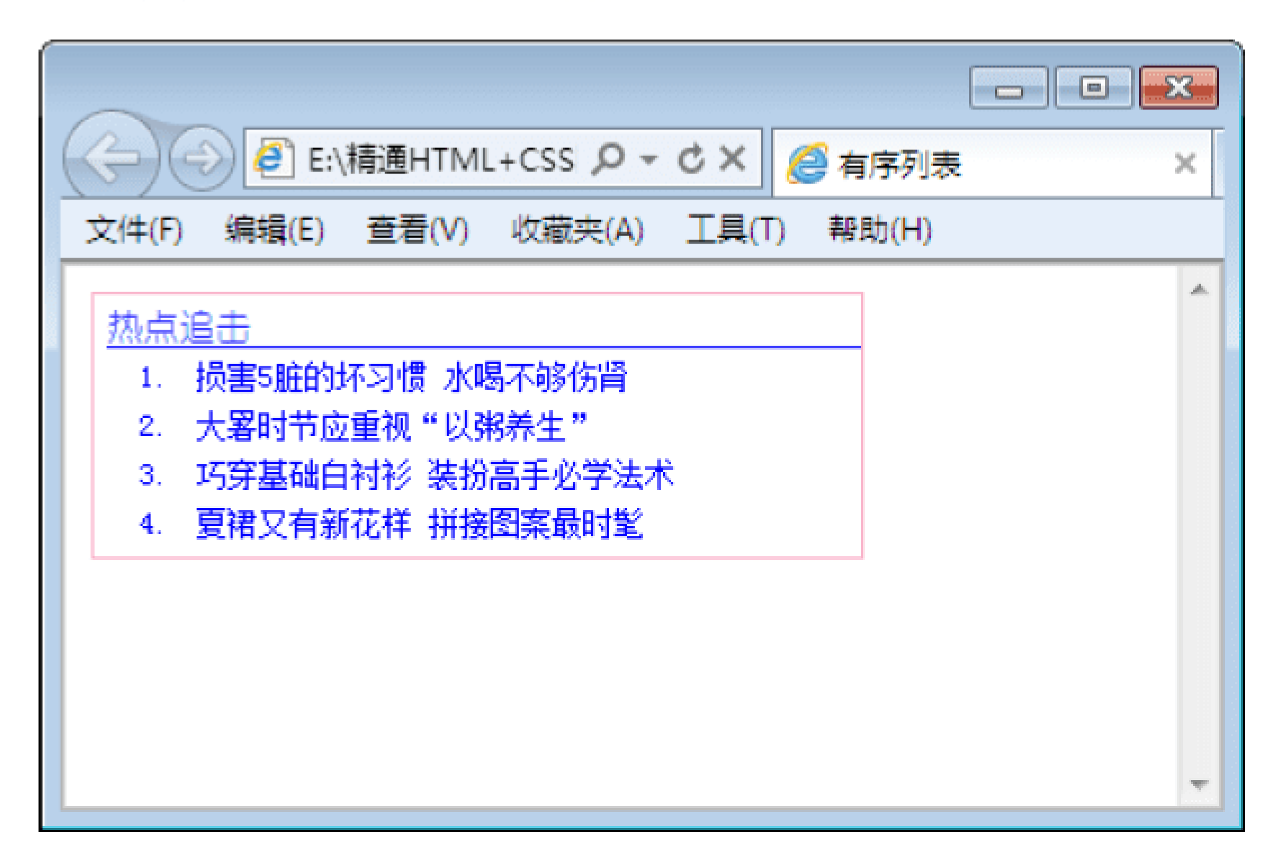


图 17-2　有序列表制作菜单

上面的代码中，使用 list-style-type: decimal 语句定义了有序列表前面的符号。严格来说，无论是<ul>标记还是<ol>标记，都可以使用相同的属性值，而且效果完全相同，即两者可以通用。此处为了方便读者学习，单独将两者分开来讲。

17.1.3　美化自定义列表

自定义列表是列表项目中比较特殊的一个列表，相对于无序和有序列表，其使用次数较少。如果要解释一系列术语，使用自定义列表就是很好的方法。引入 CSS 的一些相关属性，可以改变自定义列表显示样式。

【例 17.3】（实例文件：ch17\17.3.html）

```
<html >
<head>
<style>
*{ margin:0; padding:0;}
body{ font-size:12px; line-height:1.8; padding:10px;}
dl{clear:both; margin-bottom:5px;float:left;}
dt,dd{padding:2px 5px;float:left; border:1px solid #3366FF;width:120px;}
dd{ position:absolute; right:5px;}
```

```
h1{clear:both;font-size:14px;}
 </style>
 </head>
<body>
<h1>全部日志</h1>
<div>
<dl>
<dt><a href="#">笑</a></dt> <dd>（0/11）</dd> </dl>
 <dl> <dt><a href="#">我们还能吃什么</a></dt> <dd>（0/8）</dd> </dl>
 <dl> <dt><a href="#">太有才了</a></dt> <dd>（0/6）</dd> </dl>
 <dl> <dt><a href="#">怀念祖父</a></dt> <dd>（2/11）</dd> </dl>
 <dl> <dt><a href="#">学人二十四要</a></dt> <dd>（0/9）</dd> </dl>
 <dl> <dt><a href="#">精英教育和普及教育</a></dt> <dd>（0/2）</dd> </dl>
  <dl> <dt><a href="#">看青春剧有感</a></dt> <dd>（0/1）</dd> </dl> </div>
</body>
</html>
```

在 IE 9.0 中的浏览效果如图 17-3 所示，可以看到页面中显示一个日志导航菜单，每个选项都有蓝色边框，并且后面带有浏览次数。上面的代码中，通过使用 border 属性设置边框，使用 font 属性设置文本大小、颜色等。

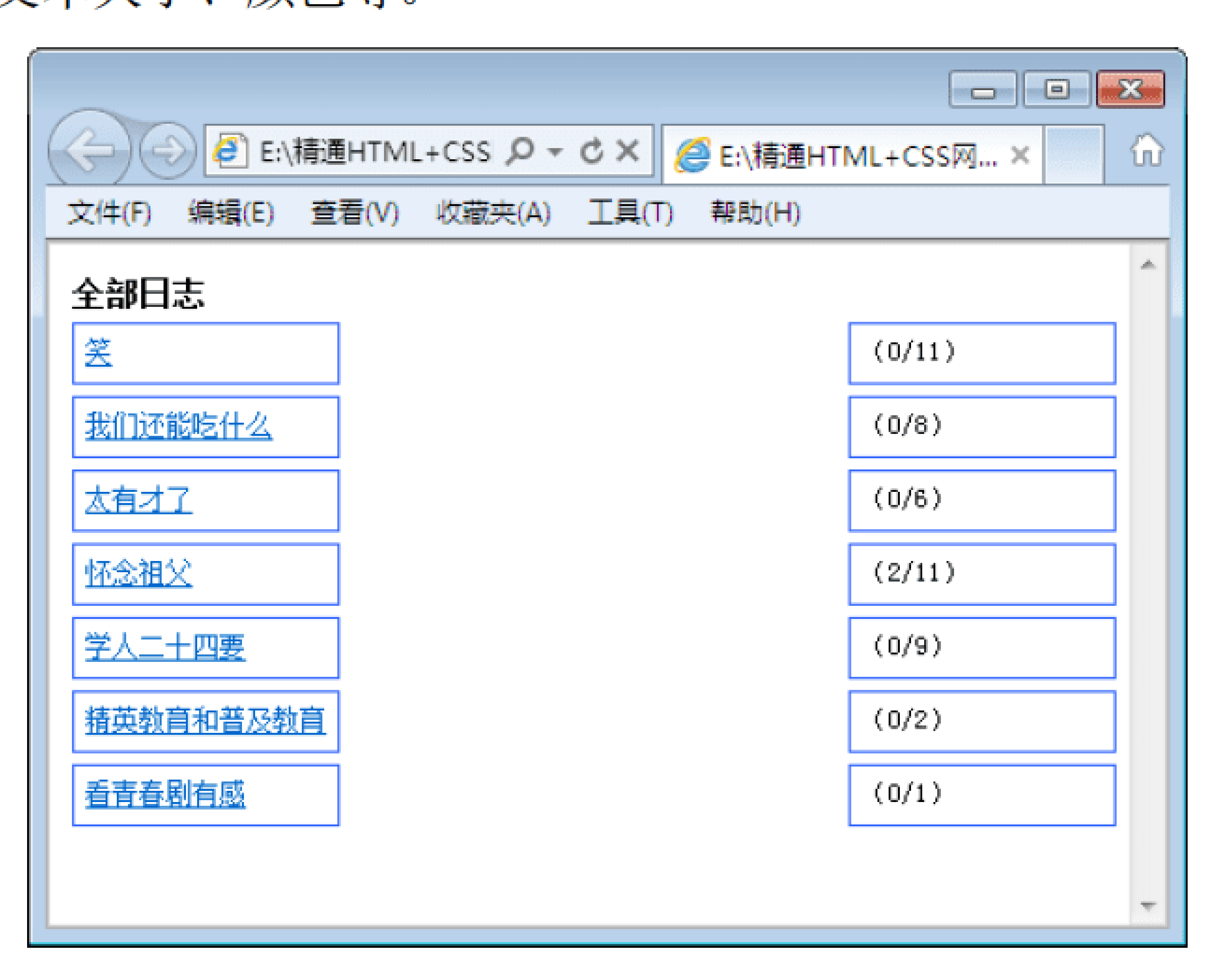


图 17-3　使用自定义列表制作导航菜单

17.1.4　图片列表

对于有序列表或无序列表，不但可以通过 list-style-type 改变选项前面的特殊符号，还可以使用 list-style-image 属性将每项前面的项目符号替换为任意的图片。

list-style-image 属性用来定义作为一个有序或无序列表项标志的图像。图像相对于列表项内容的放置位置通常使用 list-style-position 属性控制。list-style-image 的语法格式如下所示：

```
list-style-image : none | url (url)
```

上面属性值中，none 表示不指定图像，url 表示使用绝对路径和相对路径指定背景图像。

【例 17.4】（实例文件：ch17\17.4.html）

```
<html>
<head>
<title>图片符号</title>
<style>
<!--
ul{
      font-family:Arial;
      font-size:13px;
      color:#00458c;
      list-style-type:none;                              /* 不显示项目符号 */
}
li{
      list-style-image:url(6.jpg);
      padding-left:25px;                                 /* 设置图标与文字的间隔 */
      width:350px;
}
-->
</style>
    </head>
<body>
<p>健康</p>
<ul>
       <li>常吃 10 种食物 有效预防疾病更长寿 11:41 </li>
       <li>养生有禁忌:国人饮食习惯存 9 大缺陷 13:26 </li>
       <li>夏日食谱帮苦闷孕妈促食欲 14:20 </li>
       <li>夏日养生特别重要！食补养生吃这些 13:26 </li>
       <li>菜肴做法是否存在健康隐患 国人饮食习惯 9 大缺陷 13:04 </li>
</ul>
</body>
</html>
```

在 IE 9.0 中的浏览效果如图 17-4 所示，可以看到页面中显示一个导航栏，每个导航菜单前面都有一个小图标。

在上面代码中，使用 list-style-image:url(6.jpg)语句定义了列表前显示的图片，实际上还可以使用 background:url(6.jpg) no-repeat 语句完成这个效果，只不过 background 对图片大小的要求比较苛刻。

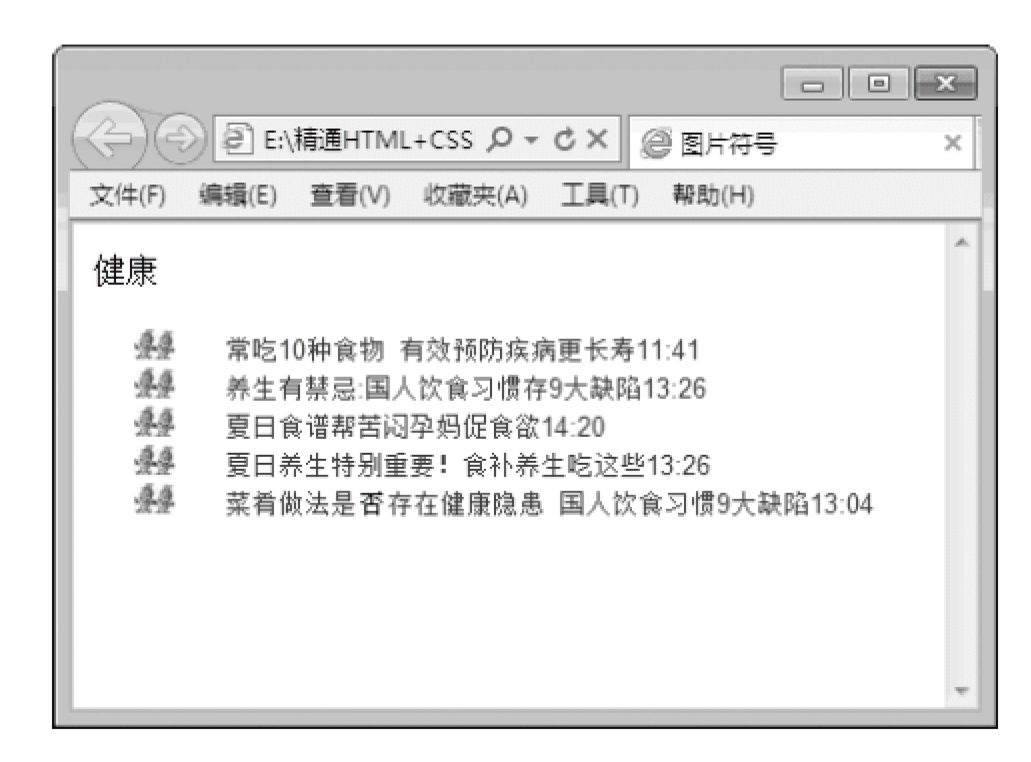


图 17-4　制作图片导航栏

17.1.5　列表缩进

使用图片作为列表符号显示时，图片通常显示在列表的外部，实际上还可以将图片列表中的文本信息对齐，从而显示另外一种效果。在 CSS 中，可以通过 list-style-position 来设置图片显示位置。

list-style-position 属性语法格式如下所示：

```
list-style-position : outside | inside
```

其属性值及含义，如表 17-3 所示。

表 17-3　列表缩进属性值

属性	说明
outside	列表项目标记放置在文本以外，且环绕文本不根据标记对齐
inside	列表项目标记放置在文本以内，且环绕文本根据标记对齐

【例 17.5】（实例文件：ch17\17.5.html）

```
<html>
<head>
<title>图片位置</title>
<style>
.list1{
    list-style-position:inside;}
.list2{
    list-style-position:outside;}
.content{
    list-style-image:url(6.jpg);
    list-style-type:none;
    font-size:24px;
}
</style>
```

```
    </head>
<body>
<ul class=content>
    <li class=list1>穴位按摩减肥法是时下较为热门的减肥方法之一，由于其操作方便、无副作用的优
    点受到众多都市女性的青睐。</li>
    <li class=list2>穴位按摩减肥法是时下较为热门的减肥方法之一，由于其操作方便、无副作用的
    优点受到众多都市女性的青睐。</li>
</ul>
</body>
</html>
```

在 IE 9.0 中的浏览效果如图 17-5 所示，可以看到一个图片列表，第一个图片列表选项中图片和文字对齐，即放在文本信息以内，第二个图片列表选项没有和文字对齐，而是放在文本信息以外。

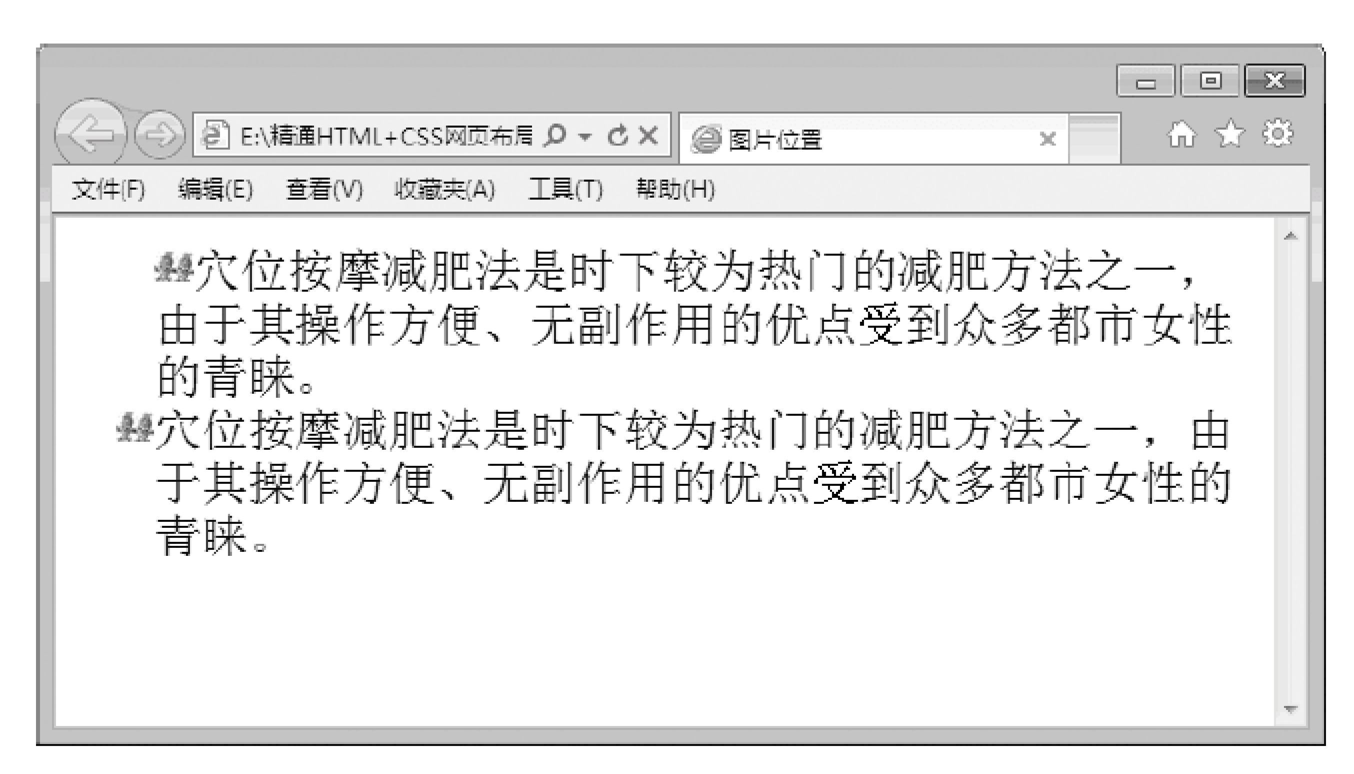


图 17-5　图片缩进

17.1.6　列表复合属性

在前面的章节中，分别使用了 list-style-type 定义列表的项目符号，使用 list-style-image 定义列表的图片符号选项，使用 list-style-position 定义图片显示位置。实际上在对项目列表操作时，可以直接使用一个复合属性 list-style，将前面的三个属性放在一起设置。

list-sytle 语法格式如下所示：

```
{ list-style: sStyle }
```

其中 sStyle 指定或接收以下值（任意次序，最多 3 个）的字符串，如表 17-4 所示。

表 17-4　list-style 常用属性

属性	说明
图像	可供 list-style-image 属性使用的图像值的任意范围
位置	可供 list-style-position 属性使用的位置值的任意范围
类型	可供 list-style-type 属性使用的类型值的任意范围

【例 17.6】（实例文件：ch17\17.6.html）

```
<html>
<head>
<title>复合属性</title>
<style>
#test1
{
        list-style:square inside url("6.jpg");
}
#test2
{
        list-style:none;
}

</style>
    </head>
<body>
<ul>
        <li id=test1>养生指南：心灵瑜珈舒解心理重负。</li>
<li id=test2>养生指南：心灵瑜珈舒解心理重负。</li>
</ul>
</body>
</html>
```

在 IE 9.0 中的浏览效果如图 17-6 所示，可以看到页面中显示两个列表选项，一个列表选项中带有图片，另一个列表中没有符号和图片显示。

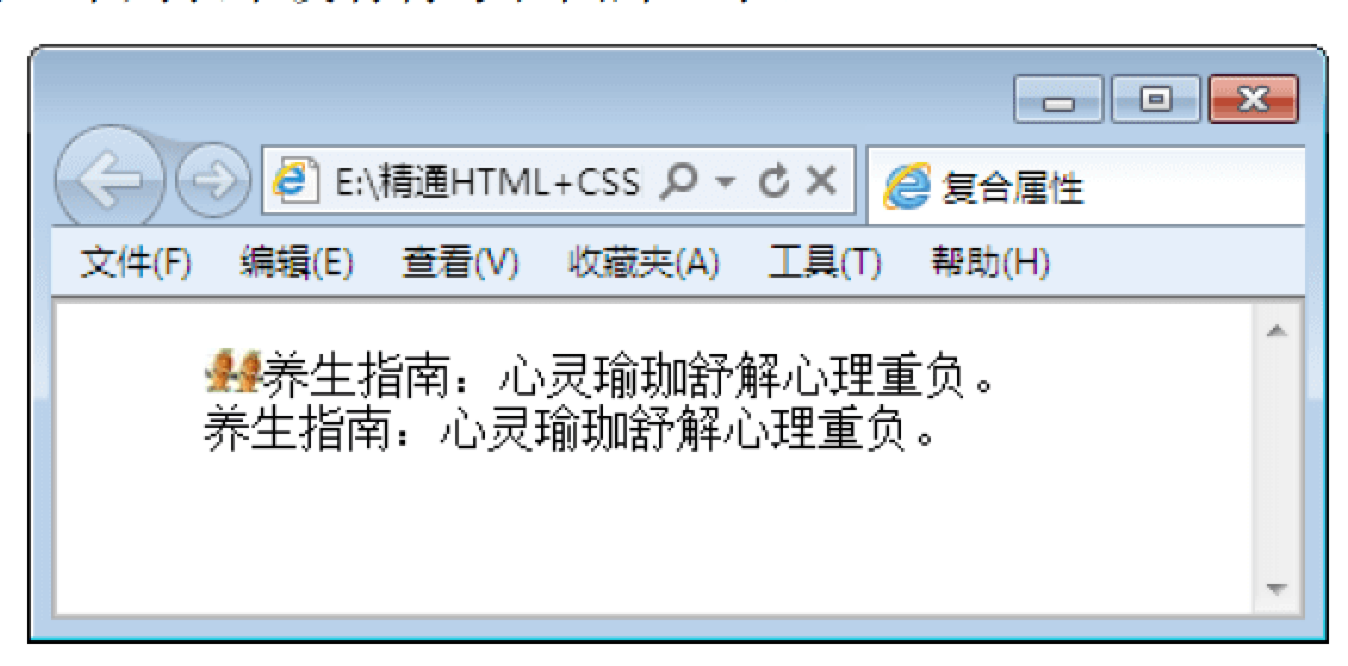


图 17-6　使用复合属性指定列表

list-style 属性是复合属性。在指定类型和图像值时，除非将图像值设置为 none 或无法显示 URL 所指向的图像，否则图像值的优先级较高。例如在上面的例子中，类 test1 同时设置了符号为方块符号和图片，但只显示了图片。

17.2　无需表格的菜单

项目列表在引入 CSS 之后，其功能和作用就大大增强了，即可以制作出各式各样的菜单和导航条。在制作导航条和菜单之前，需要将 list-style-type 值设置为 none，即去掉列表前的项目符号。

下面通过一个实例，介绍如何创建一个菜单导航条。该实例完成后，效果如图 17-7 所示。

首先创建 HTML 文档，实现一个无序列表，列表中的选项表示各个菜单。

【例 17.7】（实例文件：ch17\17.7.html）

```
<html>
<head>
<title>无需表格菜单</title>
</head>
<body>
<div>
    <ul>
        <li><a href="#">首页</a></li>
        <li><a href="#">产品</a></li>
        <li><a href="#">下载</a></li>
        <li><a href="#">购买</a></li>
        <li><a href="#">服务</a></li>
    </ul>
</div>
</body>
</html>
```

上面代码中，创建一个 div 层，在层中放置了一个 ul 无序列表，列表中的各个选项就是将来所使用的菜单。

在 IE 9.0 中浏览效果如图 17-8 所示，可以看到页面中显示了一个无序列表，每个选项带有一个实心圆。

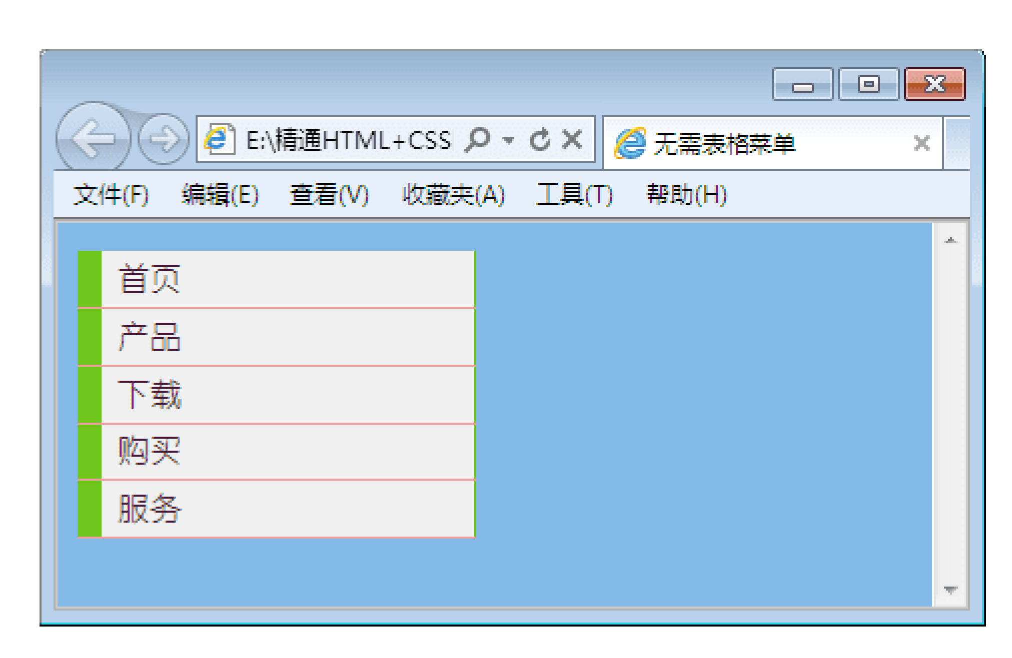


图 17-7　列表菜单

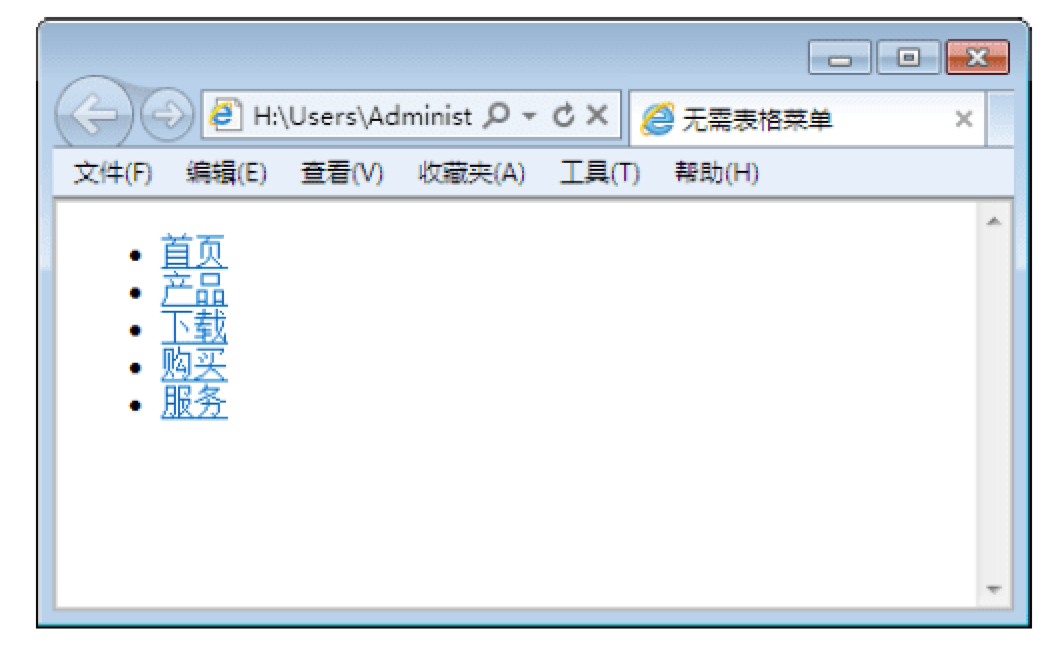


图 17-8　显示项目列表

然后可以利用 CSS 相关属性，对 HTML 中的元素进行修饰，例如 div 层、ul 列表和 body

页面。其代码如下所示：

```
<style>
<!--
body{
        background-color:#84BAE8;
}
div {
        width:200px;
        font-family:"幼圆";
}
div ul {
        list-style-type:none;
        margin:0px;
        padding:0px;
}
-->
</style>
```

在 IE 9.0 中的浏览效果如图 17-9 所示，可以看到项目列表变成一个普通的超级链接列表，无项目符号并带有下划线。

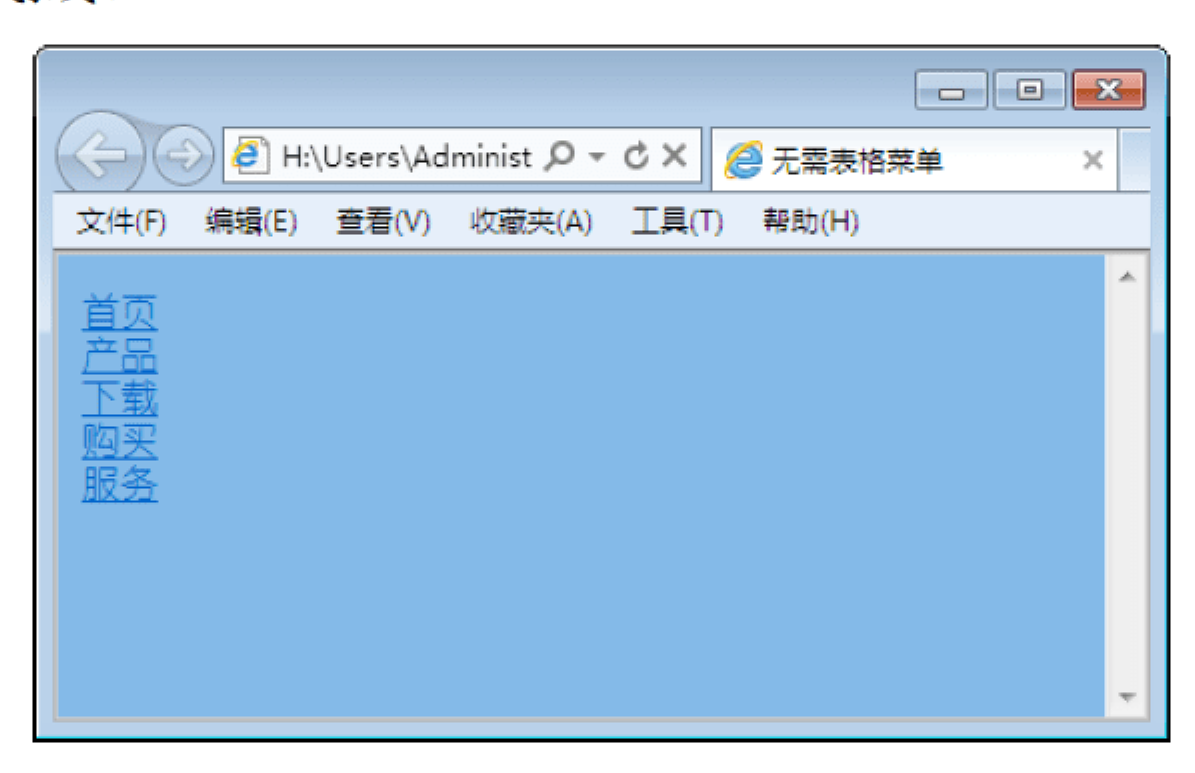


图 17-9　链接列表

接下来可以对列表中的各个选项进行修饰，例如去掉超级链接下的下划线，并增加 li 标记下的边框线，从而增强菜单的实际效果。

```
div li {
       border-bottom:1px solid #ED9F9F;
}
div li a{
       display:block;
        padding:5px 5px 5px 0.5em;
       text-decoration:none;
```

```
        border-left:12px solid #6EC61C;
        border-right:1px solid #6EC61C;
}
```

在上面代码中需要注意的语句是 display:block，此语句定义了超级链接被设置成块元素，即当鼠标进入这个区域时就被激活，而不是仅仅通过文字激活。对于 display 相关属性，会在后面章节中重点介绍。

在 IE 9.0 中的浏览效果如图 17-10 所示，可以看到每个选项中，超级链接的左方显示蓝色条，右方显示蓝色线。每个链接下方显示一个黄色边框。

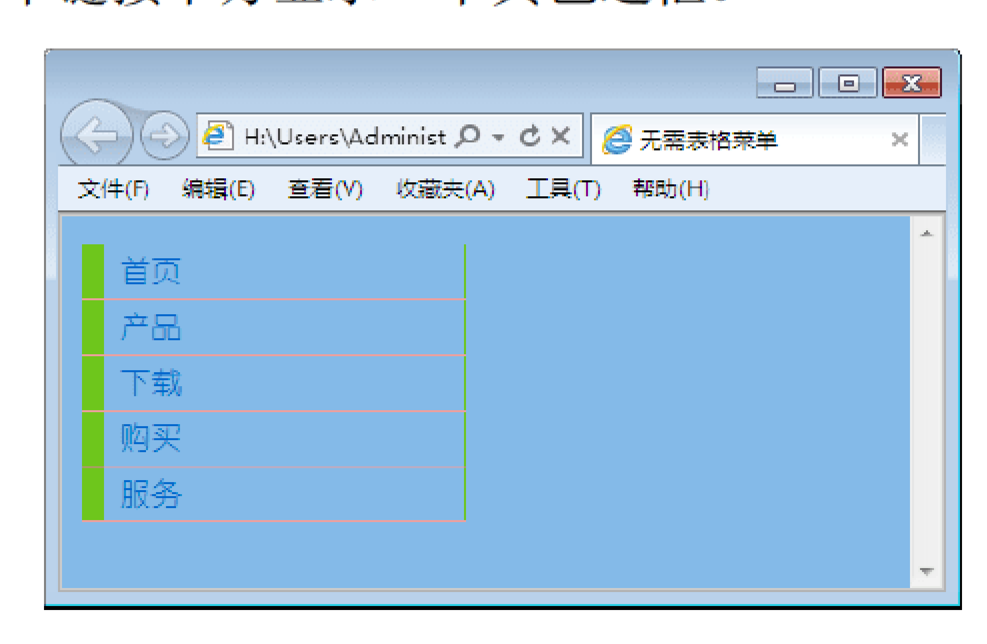


图 17-10　区块设置

当基本样式设定完成后，就可以设置导航菜单条中最常见的样式——动态菜单效果了，即当鼠标悬浮在导航菜单上时显示另外一种样式。

```
div li a:link, div li a:visited{
        background-color:#F0F0F0;
        color:#461737;
}
div li a:hover{
        background-color:#7C7C7C;
        color:#ffff00;
}
```

上面的代码设置了鼠标链接样式、访问后的样式和悬浮时的样式。

在 IE 9.0 中浏览效果如图 17-11 所示，可以看到鼠标悬浮在菜单上时，选项会显示灰色。

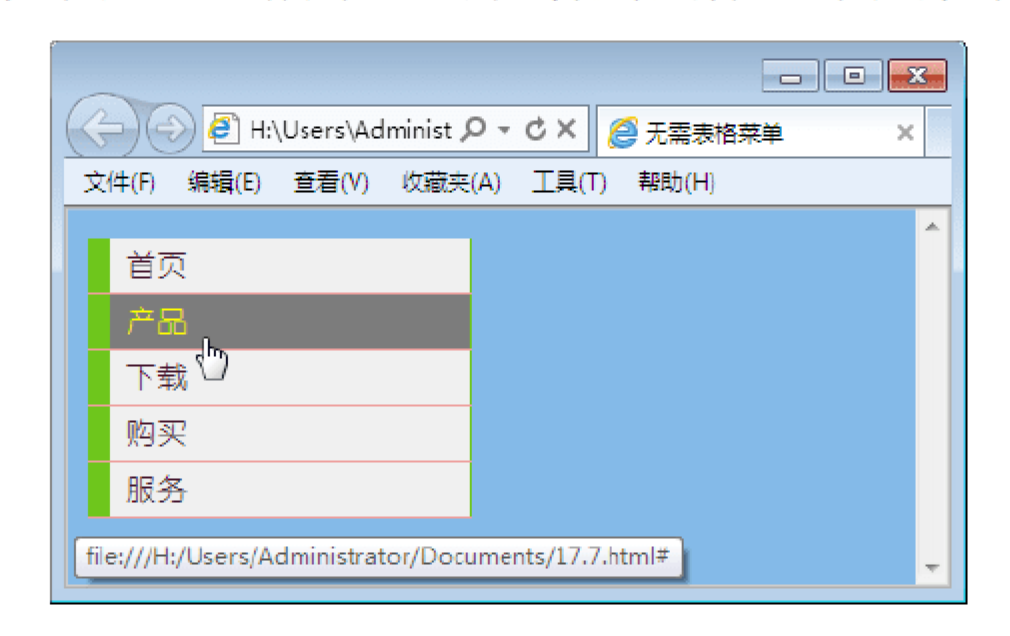


图 17-11　导航菜单

17.3　菜单的横竖转换

在实际的网页设计中，由于题材或业务需求不同，垂直导航菜单有时不能满足要求，这时就需要导航菜单水平显示。例如常见的百度首页，其导航菜单就是水平显示。通过 CSS 属性，不但可以创建垂直导航菜单，还可以创建水平导航菜单。

该实例完成后，其效果如图 17-12 所示。

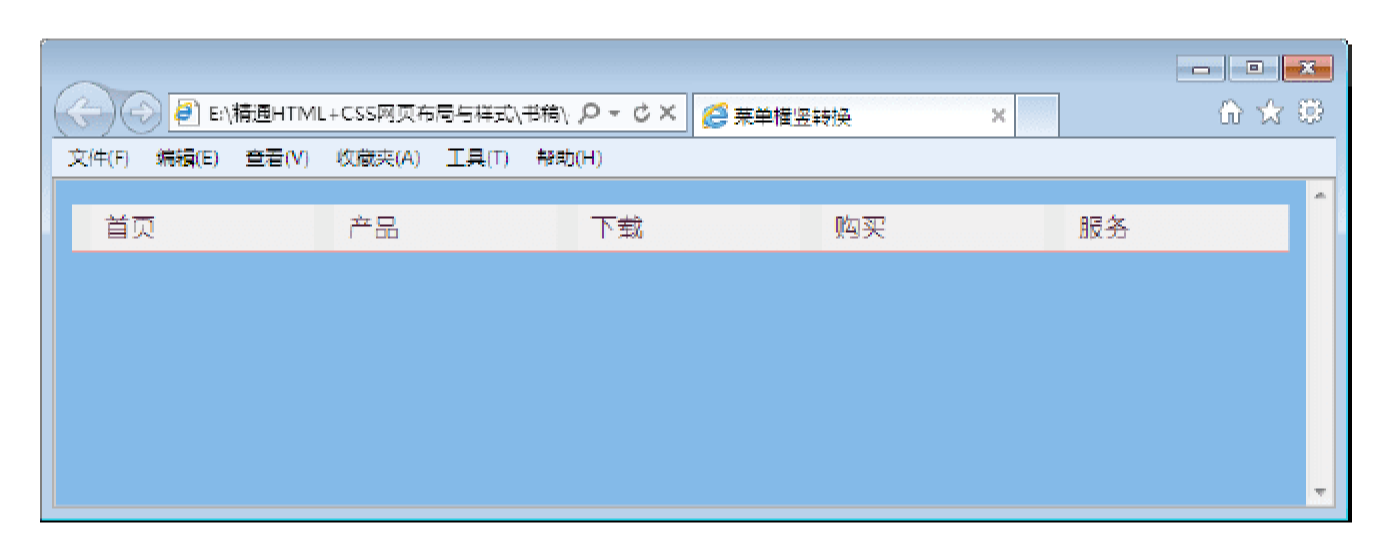


图 17-12　水平导航菜单

创建水平导航菜单和创建垂直导航菜单，前面步骤基本相同。首先需要建立 HTML 项目列表结构，将要创建的菜单项都由列表选项显示出来。然后利用 CSS 设置页面背景色；设置项目列表中的每个选项样式，去掉每个选项前面的项目符号；设置层 div 中的字体样式，但此处不需要设置 div 块。

【例 17.8】（实例文件：ch17\17.8.html）

```
<html>
<head>
<title>菜单横竖转换</title>
<style>
<!--
body{
	background-color:#84BAE8;
}
div {
	font-family:"幼圆";
}
div ul {
	list-style-type:none;
margin:0px;
	padding:0px;
}
</style>
    </head>
<body>
<div id="navigation">
```

```
<ul>
  <li><a href="#">首页</a></li>
  <li><a href="#">产品</a></li>
  <li><a href="#">下载</a></li>
  <li><a href="#">购买</a></li>
  <li><a href="#">服务</a></li>
</ul>
</div>
</body>
</html>
```

在 IE 9.0 中浏览效果如图 17-13 所示，可以看到页面中显示的是一个普通的超级链接列表，和例 17.7 中显示的基本一样。

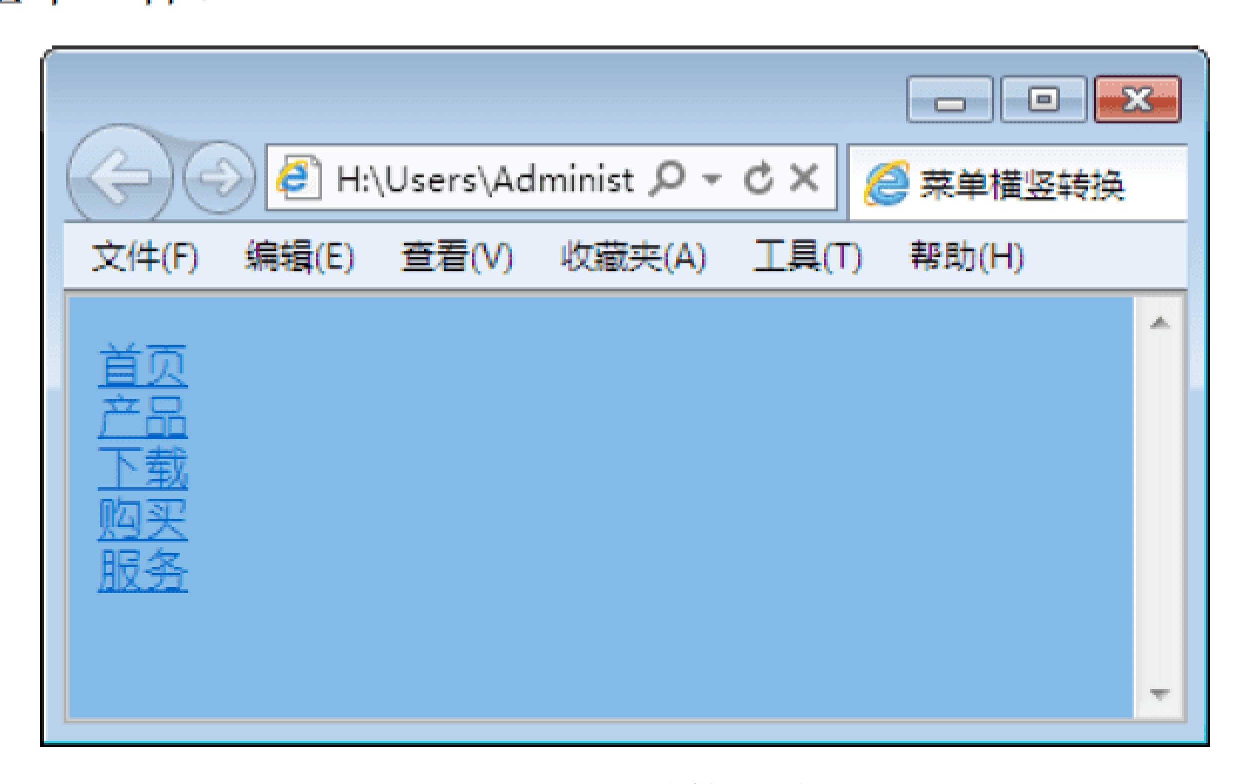


图 17-13　链接列表

现在是垂直显示导航菜单，需要利用 CSS 属性 float 将其设置为水平显示，并设置选项 li 和超级链接的基本样式，代码如下所示。

```
div li {
    border-bottom:1px solid #ED9F9F;
    float:left;
    width:150px;
}
div li a{
    display:block;
    padding:5px 5px 5px 0.5em;
    text-decoration:none;
    border-left:12px solid #EBEBEB;
    border-right:1px solid #EBEBEB;
}
```

当 float 属性值为 left 时，导航栏为水平显示。其他设置基本和例 17.7 相同。在 IE 9.0 中浏览效果如图 17-14 所示，可以看到各个链接选项水平排列在当前页面之上。

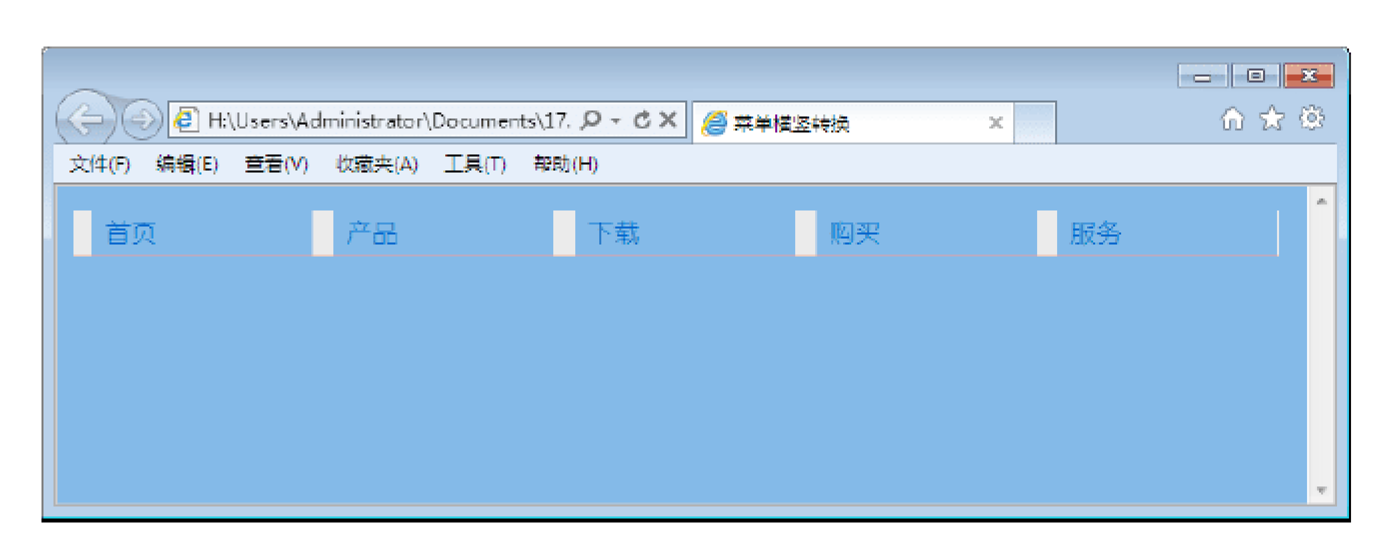


图 17-14 列表水平显示

下面设置超级链接<a>的样式，和前面一样，也设置了鼠标动态效果。代码如下所示。

```
div li a:link, div li a:visited{
        background-color:#F0F0F0;
        color:#461737;
}
div li a:hover{
        background-color:#7C7C7C;
        color:#ffff00;
}
```

在 IE 9.0 中浏览效果如图 17-15 所示，可以看到当鼠标放到菜单之上时，会变换为另一种样式。

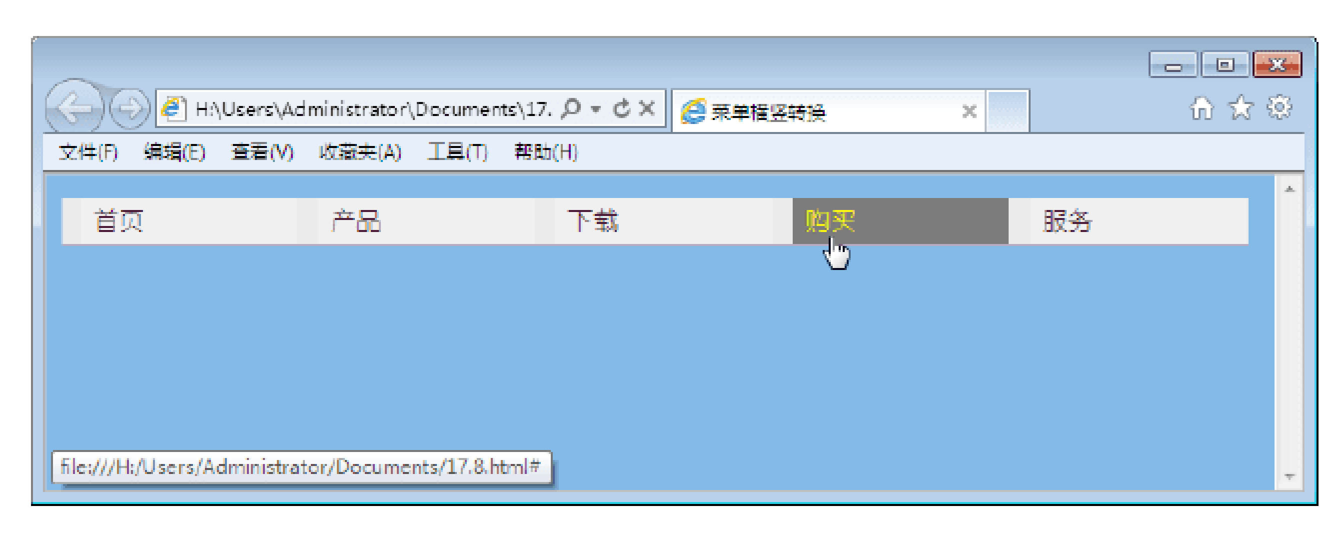


图 17-15 水平菜单显示

17.4 综合实例 1——段落模拟列表效果

CSS 是伟大的，通过它可以变换不同的样式。可以让列表代替 table 表格制作出表格，同样也可以让一个段落模拟项目列表。本实例将利用前面介绍的 CSS 知识，将段落变换为一个列表。具体步骤如下所示。

01 分析需求。

要实现这个实例，需要创建一个 div 层，用于包含各个不同的段落，以方便设置整体布局。该实例实现后，效果如图 17-16 所示。

02 创建 HTML，实现基本段落。

从上面的分析可以看出，HTML 中需要包含一个 div 层，几个段落。其代码如下所示。

```
<html>
<head>
<title>模拟列表</title>
</head>
<body>
<div class="big">
  <p  class="one">• 英国两头小猪遭主人弃养 游泳千米找回家(图)新浪网  7 小时前    </p>
  <p> • 7 岁女孩参加小区游泳培训班溺亡 警方调查腾讯网  8 小时前   </p>
  <p  class="one"> • 总局为姚明举办退役仪式 送上最高荣誉和敬意搜狐网  21 小时前  </p>
  <p> • 姚明获总局多项至高殊荣 大姚告别礼哽咽离去腾讯网  12 小时前  </p>
  <p class="one"> • 大学生五人制联赛官网上线 推动高校足球事业腾讯网  6 小时前 </p>
</div>
</body>
</html>
```

在 IE 9.0 中浏览效果如图 17-17 所示，可以看到页面中显示 5 个段落，每个段落前面都使用特殊符号“ • ”引领每一行。

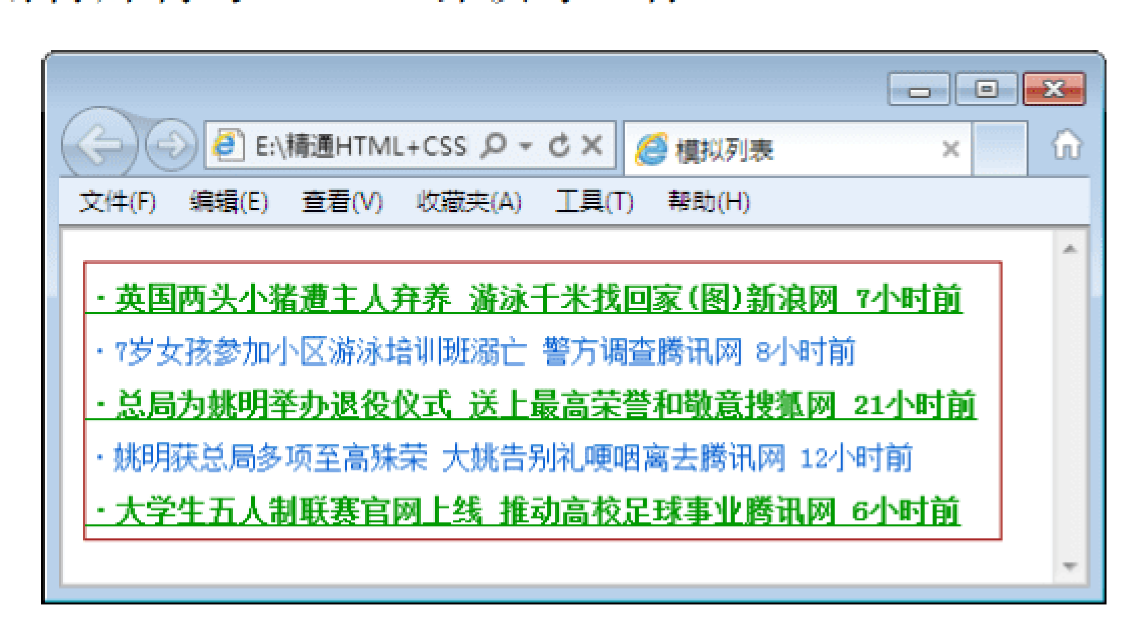


图 17-16　段落模拟列表

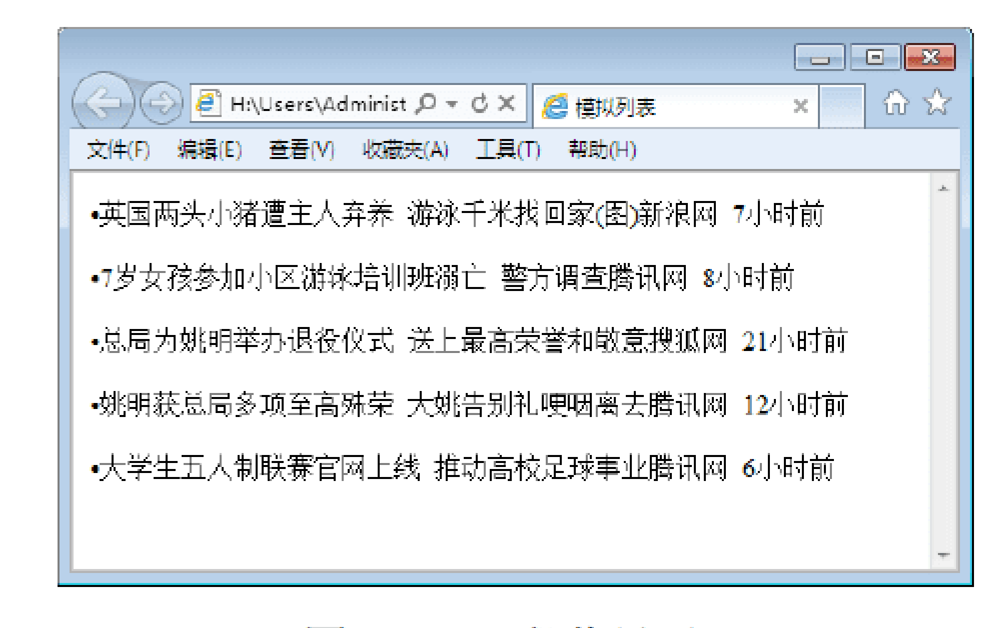


图 17-17　段落显示

03 添加 CSS 代码，修饰整体 div 层。

```
<style>
.big {
    width:450px;
    border:#990000 1px solid;
}
</style>
```

此处创建了一个类选择器，其属性定义了层的宽度，层带有边框，以直线形式显示。

在 IE 9.0 中浏览效果如图 17-18 所示，可以看到段落周围显示了一个矩形区域，其边框显示为黄色。

04 添加 CSS 代码，修饰段落属性。

```
p {
        margin:10px 0 5px 0;
        font-size:14px;
        color:#025BD1;
}
.one {
        text-decoration:underline;
        font-weight:800;
        color:#009900;
}
```

上面代码定义了段落 p 的通用属性，即字体大小和颜色。使用类选择器定义了特殊属性，带有下划线，具有不同的颜色。

在 IE 9.0 中浏览效果如图 17-19 所示，可以看到与图 17-18 相比，其字体颜色发生了变化，并带有下划线。

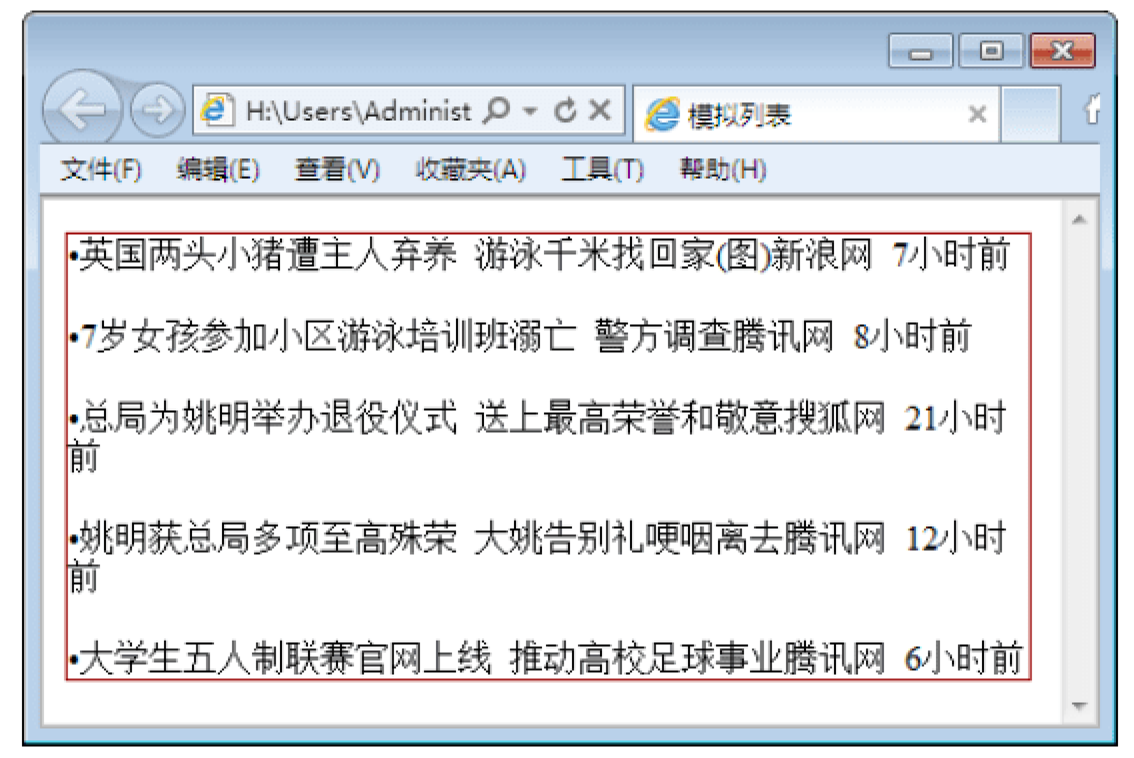


图 17-18 设置 div 层

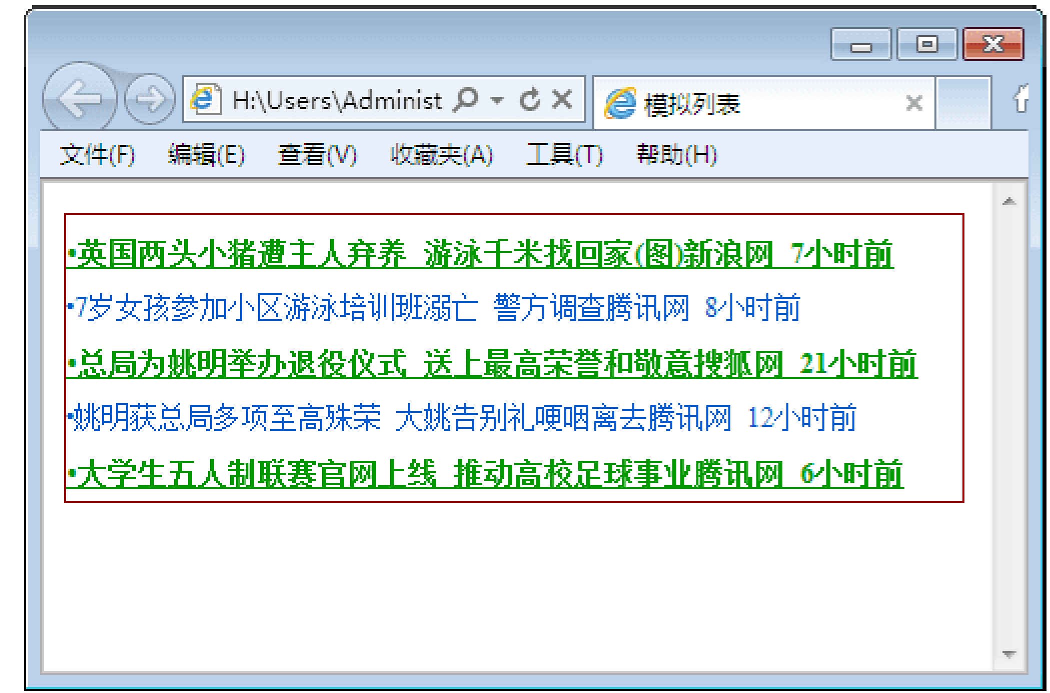


图 17-19 修饰段落属性

17.5 综合实例 2——模拟 soso 导航栏

作为一个搜索引擎网站，搜搜知名度越来越高了。打开其首页，可以看到一个水平导航菜单，通过这个导航可以搜索不同类别的内容。本实例将结合本章学习的导航菜单知识，轻松实现搜搜导航栏。

本实例的宗旨是实现水平导航栏，其他方面没有过多定义。要实现该实例，具体步骤如下所示。

01 分析需求。

该实例需要包含 3 个部分，第一部分是 soso 图标，第二部分是水平菜单导航栏，也是本实例的重点，第三部分是表单部分，包含输入框和按钮。该实例实现后，其实际效果如图 17-20 所示。

02 创建 HTML 网页，实现基本 HTML 元素。

对于本实例，需要利用 HTML 标记实现搜搜图标、导航的项目列表、下方的搜索输入框和按钮等。其代码如下所示。

```
<html>
<head>
<title>搜搜</title>
</head>
<body>
<center><br><img src="logo_index.png"><br><br><br><br>
<div>
<ul>
	<li id=h></li>
	<li><a href="#">网页</a></li>
	<li > <a href="#">图片</a></li>
	<li> <a href="#">视频</a></li>
	<li><a href="#">音乐</a></li>
	<li><a href="#">搜吧</a></li>
	<li><a href="#">问问</a></li>
	<li><a href="#">团购</a></li>
	<li><a href="#">新闻</a></li>
	<li><a href="#">地图</a></li>
	<li id="more"><a href="#">更 多 &gt;&gt;</a></li>
</ul>
</div>
<p style="height:44px;"> </p>
<div id=s>
<form action="/q?" id="flpage" name="flpage">
	<input type="text" value="" size=50px;/>
	<input type="submit" value="搜搜">
</form>
</div>
</center>
</body>
</html>
```

在 IE 9.0 中浏览效果如图 17-21 所示，可以看到页面中显示了一个图片，即搜搜图标，中间显示了项目列表，每个选项都是超级链接。下方是一个表单，包含输入框和按钮。



图 17-20　模拟搜搜导航栏



图 17-21　页面框架

03 添加 CSS 代码，修饰项目列表。

框架出来之后，就可以修改项目列表的相关样式了，即使列表水平显示，同时定义整个 div 层属性，例如设置背景色、宽度、底部边框和字体大小等。代码如下所示。

```
p{ margin:0px; padding:0px;}
#div{
    margin:0px auto;
    font-size:12px;
    padding:0px;
    border-bottom:1px solid #00c;
    background:#eee;
    width:800px;height:18px;
}
div li{
    float:left;
    list-style-type:none;
    margin:0px;padding:0px;
    width:40px;
}
```

上面代码中，float 属性设置菜单栏水平显示，list-style-type 属性设置了列表不显示项目符号。在 IE 9.0 中浏览效果如图 17-22 所示，可以看到页面整体效果和搜搜首页比较相似，下面就可以在细节上做进一步修改了。

04 添加 CSS 代码，修饰超级链接。

```
div li a{
    display:block;
```

```
        text-decoration:underline;
        padding:4px 0px 0px 0px;
        margin:0px;
         font-size:13px;
}
div li a:link, div li a:visited{
        color:#004276;

}
```

上面的代码设置了超级链接，即导航栏中菜单选项中的相关属性，例如超级链接以块显示、文本带有下划线、字体大小为 13 像素，并设定了鼠标访问超级链接后的颜色。

在 IE 9.0 中浏览效果如图 17-23 所示，可以看到字体颜色发生了改变，字号也变小了。





图 17-22　水平菜单栏



图 17-23　设置菜单样式

05 添加 CSS 代码，定义对齐方式和表单样式。

```
div li#h{width:180px;height:18px;}
div li#more{width:85px;height:18px;}
#s{
      background-color:#006EB8;
      width:430px;
}
```

上述代码中，h 定义了水平菜单最前方空间的大小，more 定义了更多的长度和宽度，s 定义了表单背景色和宽度。

在 IE 9.0 中浏览效果如图 17-24 所示，可以看到水平导航栏和表单对齐，表单背景色为蓝色。

06 添加 CSS 代码，修饰访问时的样式。

```
<a href="#"   style="text-decoration:none;color:#020202;font-size:14px;">网页</a>
```

此代码段设置了被访问时的默认样式。在 IE 9.0 中浏览效果如图 17-25 所示，可以看到“网页”菜单选项被访问时，文字颜色为黑色，不带有下划线。



图 17-24　设置菜单选项大小



图 17-25　搜搜最终效果

17.6　专家解惑

1．使用项目列表和 table 表格制作表单时，项目列表有哪些优势？

采用项目列表制作水平菜单时，如果没有设置<ul>标记的宽度 width 属性，那么当浏览器的宽度缩小时，菜单会自动换行。这是采用<table>标记制作菜单所无法实现的，所以项目列表经常被使用，以实现各种变换效果。

2．使用 IE 浏览器打开一个项目列表，设定的项目符号为何没有出现？

IE 浏览器对项目列表的符号支持不是太好，只支持一部分项目符号，这时可以采用 IE 9.0 浏览器。IE 9.0 浏览器对项目列表符号支持力度比较大。

3．使用 url 引入图像时，加引号好，还是不加引号好？

不加引号好。需要将带有引号的修改为不带引号的。例如将 background:url("xxx.gif")改为 background:url(xxx.gif)。

因为对于部分浏览器而言加引号反而会引起错误。

第 18 章

CSS滤镜的应用

随着网页设计技术的发展，人们已经不满足于单调地展示页面布局并显示文本，而是希望在页面中能够加入一些多媒体特效而使页面丰富起来。使用滤镜能够实现这些需求，它能够产生各种各样的文字或图片特效，从而大大提高页面的吸引力。

18.1　滤镜概述

CSS 滤镜是 IE 9.0 浏览器厂商为了增加浏览器功能和竞争力，而独自推出的一种网页特效。CSS 滤镜不是浏览器插件，也不符合 CSS 标准。由于 IE 9.0 浏览器应用比较广泛，所以这里对其进行了介绍。

从 Internet Explorer 4.0 开始，IE 浏览器便开始支持多媒体滤镜特效，允许使用简单的代码就能对文本和图片进行处理，如模糊、彩色投影、火焰效果、图片倒置、色彩渐变。风吹效果和光晕效果等。如果把滤镜和渐变结合运用到网页脚本语言中，就可以建立一个动态交互的网页。

CSS 的滤镜属性的标识符是 filter，语法格式如下。

```
filter:filtername(parameters)
```

filtername 是滤镜名称，例如 Alpha、blur、chroma 和 DropShadow 等，parameters 指定了滤镜中的参数，通过这些参数才能够决定滤镜的显示效果。表 18-1 列出常用滤镜的名称及其效果。

表 18-1　CSS 滤镜

滤镜名称	效果
Alpha	设置透明度
BlendTrans	实现图像之间的淡入和淡出的效果
Blur	建立模糊效果
Chroma	设置对象中指定的颜色为透明色
DropShadow	建立阴影效果
FlipH	将元素水平翻转
FlipV	将元素垂直翻转

（续表）

滤镜名称	效果
Glow	建立外发光效果
Gray	灰度显示图像，即显示为黑白图像
Invert	图像反相，包括色彩、饱和度和亮度值，类似底片效果
Light	设置光源效果
Mask	建立透明遮罩
RevealTrans	建立切换效果
Shadow	建立另一种阴影效果
Wave	波纹效果
Xray	显现图片的轮廓，类似于 X 光片效果

18.2 CSS 的基本滤镜

CSS 滤镜可以分为基本滤镜和高级滤镜，基本滤镜是直接作用于 HTML 对象上便能立即生效的滤镜。

18.2.1 通道（Alpha）滤镜

Alpha 滤镜能实现针对图片文字元素的“透明”效果，这种透明效果是通过“把一个目标元素和背景混合”来实现的，混合程度可以由用户指定数值来控制。通过指定坐标，可以指定点、线和面的透明度。如果将 Alpha 滤镜与网页脚本语言结合，并适当地设置其参数，就能使图像显示淡入淡出的效果。

Alpha 滤镜的语法格式如下。

```
{filter : Alpha ( enabled=bEnabled, style=iStyle, opacity=iOpacity, finishOpacity=iFinishOpacity,
        startx=iPercent, starty=iPercent, finishx=iPercent, finishy=iPercent )}
```

各参数如表 18-2 所示。

表 18-2 Alpha 滤镜参数及说明

参数	说明
enabled	设置滤镜是否激活
style	设置透明渐变的样式，也就是渐变显示的形状，取值为 0~3。0 表示无渐变，1 表示线形渐变，2 表示圆形渐变，3 表示矩形渐变
opacity	设置透明度，值范围是 0~100。0 表示完全透明，100 表示完全不透明
finishOpacity	设置结束时透明度，值范围也是 0~100
startx	设置透明渐变开始点的水平坐标（即 x 坐标）
Starty	设置透明渐变开始点的垂直坐标（即 y 坐标）

（续表）

参数	说明
finishx	设置透明渐变结束点的水平坐标
finishy	设置透明渐变结束点的垂直坐标

【例 18.1】（实例文件：ch18\18.1.html）

```
<html>
<head>
    <title>Alpha 滤镜</title>
</head>
<body>
    原始图<img src="baimd.jpg" style="width:200px;height:120px;">
    style=0<img src="baimd.jpg" style="width:200px;height:120px;filter : Alpha(opacity=80,style=0)" >
    style=2<img src="baimd.jpg" style="width:200px;height:120px;filter : Alpha(opacity=60,style=2)" >
    style=3 <img src="baimd.jpg" style="width:200px;height:120px;filter : Alpha(opacity=40, style=3)" >
  </body>
</html>
```

在 IE 9.0 中浏览效果如图 18-1 所示，可以看到页面中显示了 4 张图片，其透明度依次减弱。

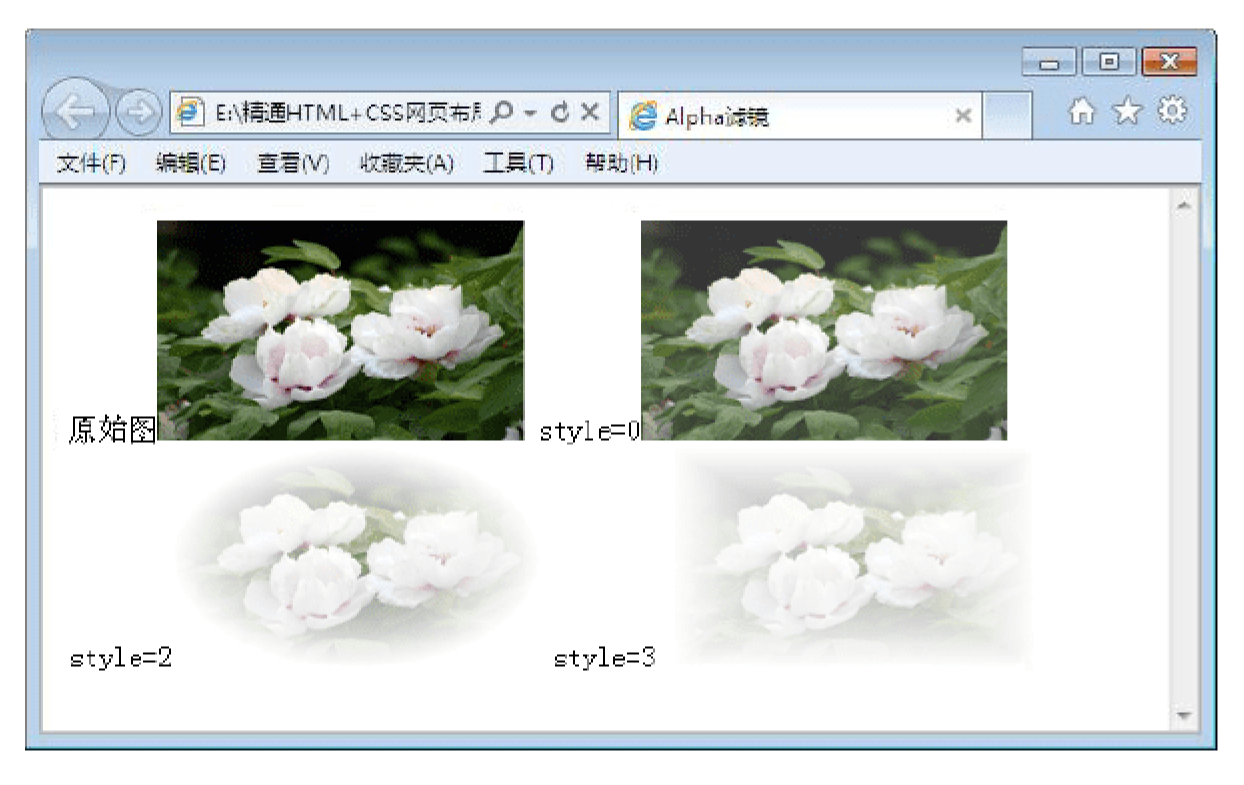


图 18-1　实现图片滤镜效果

在使用 Alpha 滤镜时要注意以下两点。

- 由于 Alpha 滤镜使当前元素部分透明，该元素下层的内容的颜色对整个效果起着重要作用，因此颜色的合理搭配相当重要。
- 透明度的大小要根据具体情况仔细调整，取一个最佳值。

Alpha 滤镜不但能应用于图片，还可以应用于文字。

【例 18.2】（实例文件：ch18\18.2.html）

```
<html>
<head>
```

```
<title>Alpha 滤镜</title>
<style type="text/css">
<!--
p{
      color:yellow;
      font-weight:bolder;
      font-size:25pt;
      width:100%
}
-->
</style>
</head>
<body style="background-color:Black">
      <div >
         <p>Alpha 滤镜</p>
         <p style="filter:alpha(opacity=80, style=1)">透明效果</p>
         <p style="filter:alpha(opacity=60, style=2)">透明效果</p>
         <p style="filter:alpha(opacity=40, style=3)">透明效果</p>
      </div>
   </body>
</html>
```

在 IE 9.0 中浏览效果如图 18-2 所示，可以看到页面中显示 4 行文字，其透明度依次减弱。

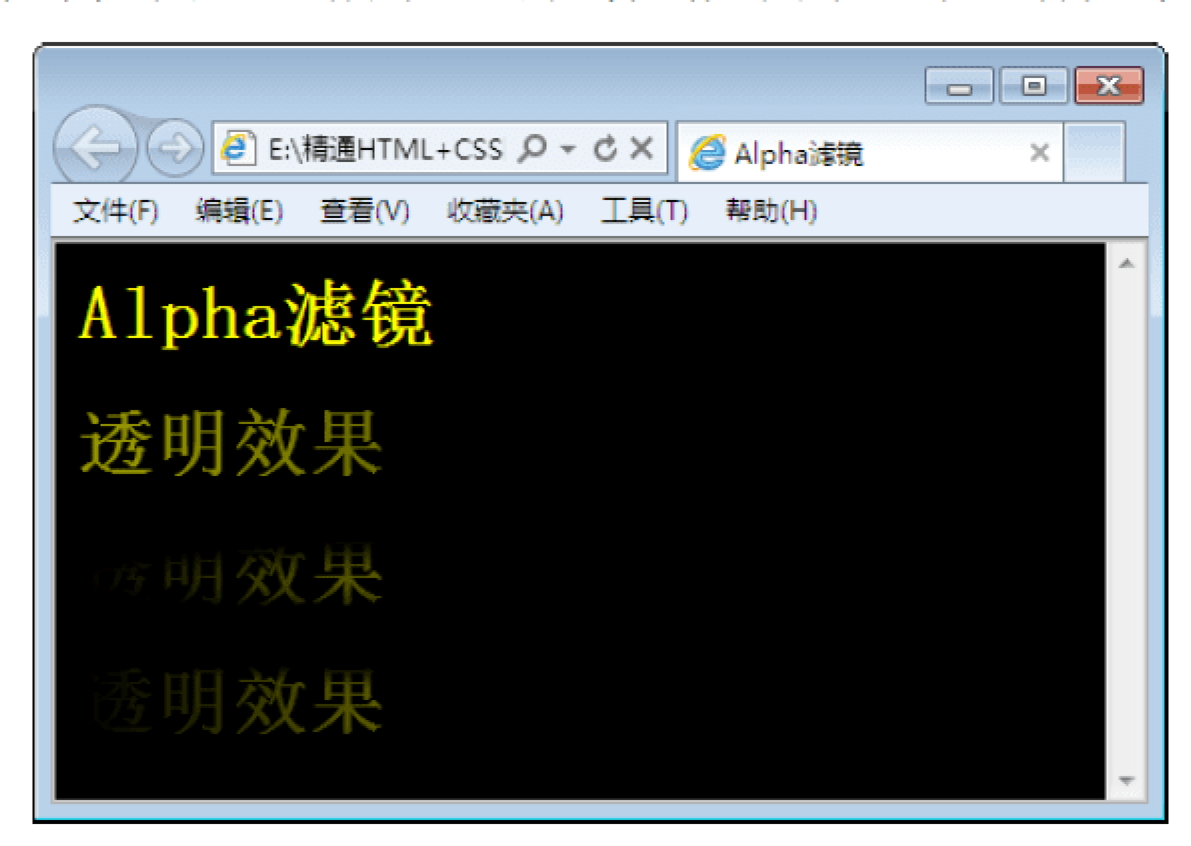


图 18-2　文字透明效果

18.2.2　Blur 滤镜

Blur 滤镜实现页面模糊效果，即在一个方向上的运动模糊。如果应用得当，就可以产生高速移动的动感效果。

Blur 滤镜的语法格式如下。

```
{filter : Blur ( enabled=bEnabled , add=iadd , direction=idirection ,
```

```
strength=fstrength )}
```

其参数如表 18-3 所示。

表 18-3　Blur 滤镜参数及说明

参数	说明
enabled	设置滤镜是否激活
add	指定图片是否改变成模糊效果。这是个布尔参数，有效值为 True 或 False。True 是默认值，表示应用模糊效果，False 则表示不应用
direction	设定模糊方向。模糊的效果是按顺时针方向起作用的，取值范围为 0~360 度，45 度为一个间隔。有 8 个方向值：0 表示零度，代表向上方向。45 表示右上，90 表示向右，135 表示右下，180 表示向下。225 表示左下，270 表示向左，315 表示左上
strength	指定模糊半径大小，单位是像素，默认值为 5，取值范围为自然数，该取值决定了模糊效果的延伸范围

【例 18.3】（实例文件：ch18\18.3.html）

```
<html>
<head>
<title>模糊 Blur</title>
<style>
img{
    height:180px;
}
div.div2 { width:400px;filter:blur(add=true,direction=90,strength=50) }
</style>
</head>
<body>
    原始图<img src="baihua.jpg">
    add=true<img src="baihua.jpg" style="filter:Blur(add=true,direction=225,strength=20)">
    add=false<img src="baihua.jpg" style="filter:Blur(add=false,direction=225,strength=20)">
    <div class="div2">
    <p style="font-size: 30pt; font-weight: bold; color:DarkBlue">
    Blur 滤镜</p>
    </div>
</body>
</html>
```

在 IE 9.0 中浏览效果如图 18-3 所示，可以看到两张模糊图片是在一定方向上发生的模糊。下方的文字也发生了模糊，具有吹风的效果。

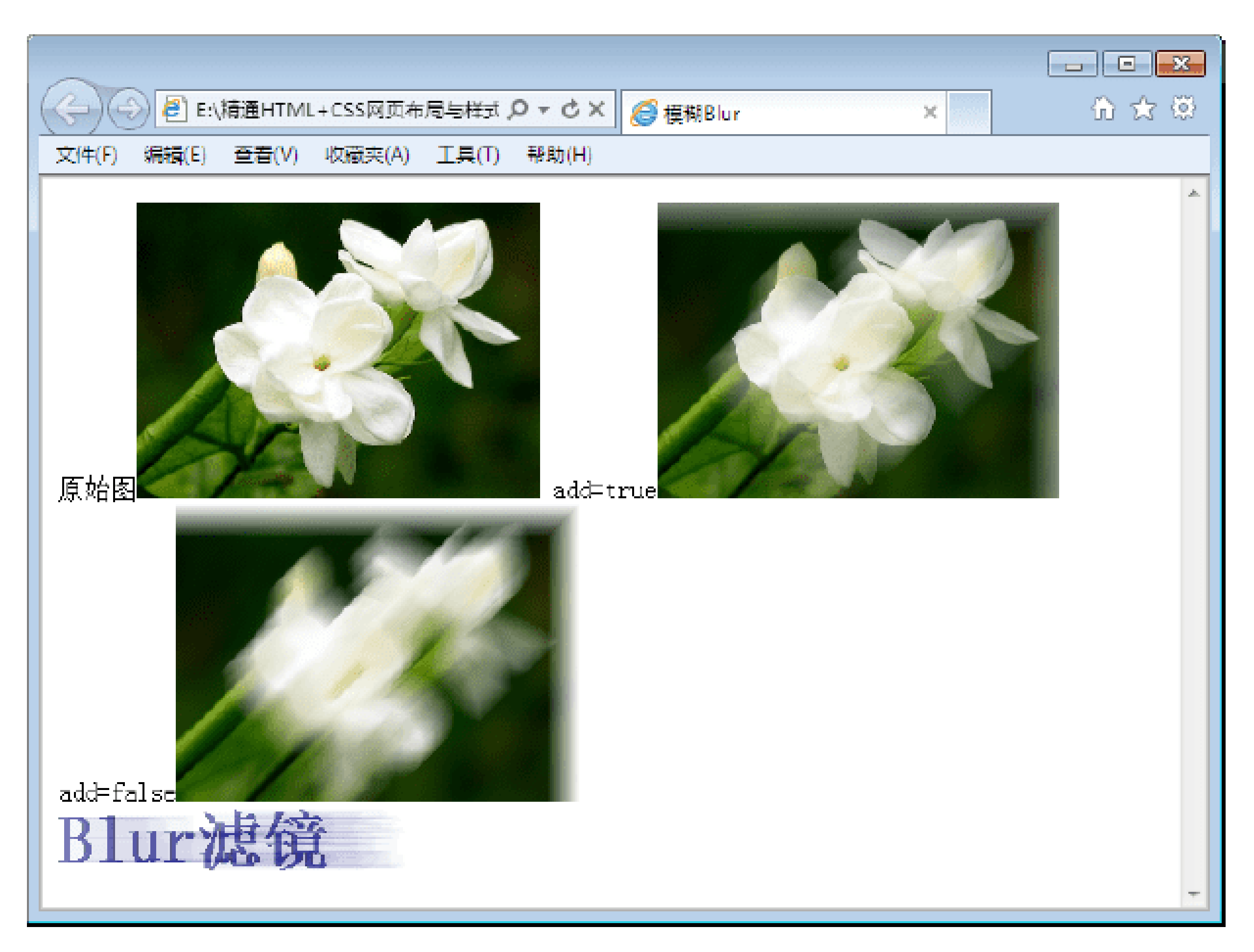


图 18-3　模糊效果

18.2.3　Chroma 滤镜

Chroma 滤镜可以设置 HTML 对象中指定的颜色为透明色。其语法格式如下。

```
{filter : Chroma(enabled=bEnabled , color=sColor)}
```

其中，color 参数设置要变为透明色的颜色。

【例 18.4】（实例文件：ch18\18.4.html）

```
<html>
<head>
	<title>Chroma 滤镜</title>
<style>
<!--
div{position:absolute;top:70;letf:40; filter:Chroma(color=blue)}
p{font-size:30pt; font-weight:bold; color:blue}
-->
</style>
</head>
<body>
	<p>Chroma 滤镜效果</p>
	<div>
		<p>Chroma 滤镜效果</p>
	</div>
</body>
</html>
```

在 IE 9.0 中浏览效果如图 18-4 所示，可以看到第二行文字的某些笔画丢失。拖动鼠标选择过滤颜色后的文字，便可以查看过滤掉颜色的文字。

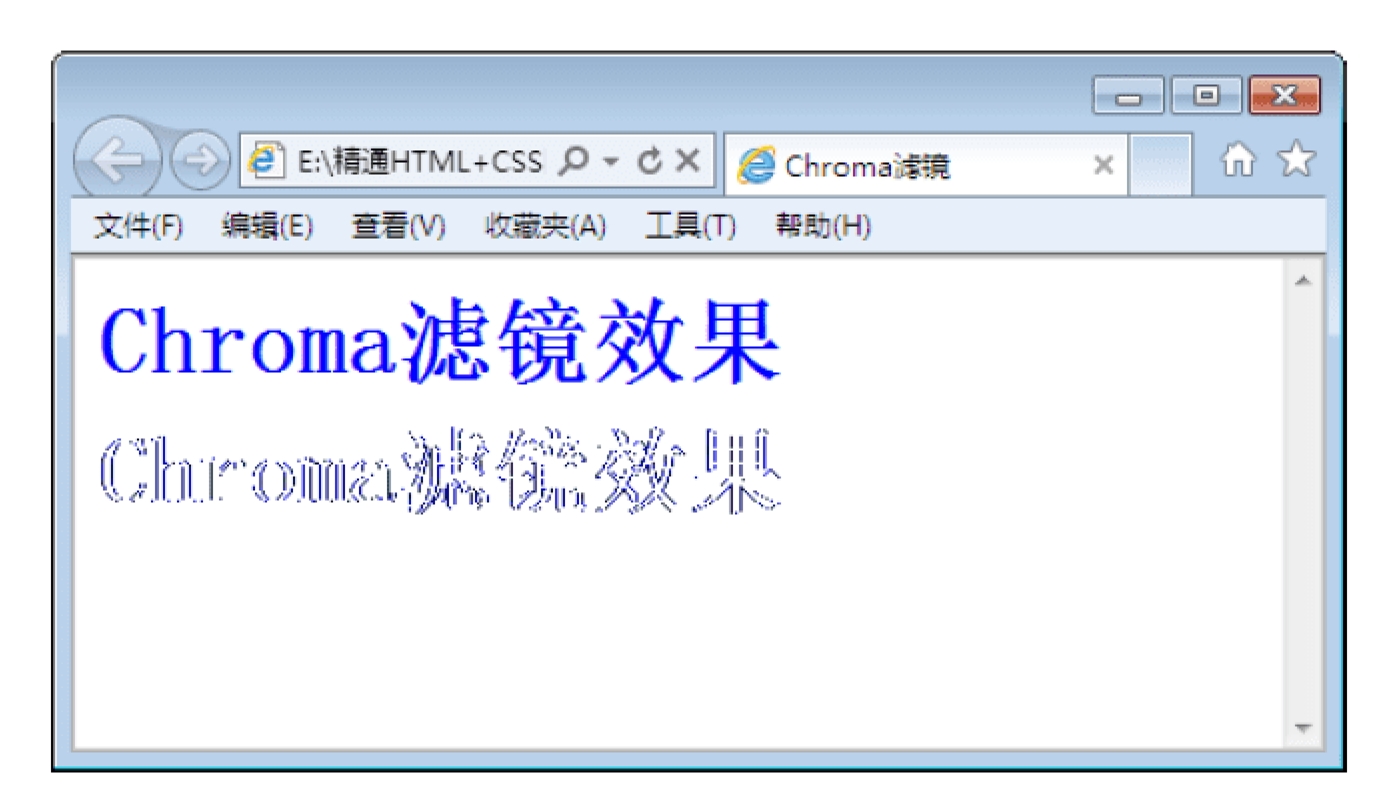


图 18-4　Chroma 滤镜效果

Chroma 滤镜一般应用于文字特效，而且对于有些格式的图片也是不适用的。例如，JPEG 格式的图片是一种已经减色和压缩处理的图片，所以要设置其中某种颜色为透明十分困难。

18.2.4　DropShadow 滤镜

阴影效果在实际的文字和图片中非常实用，IE 9.0 通过 DropShadow 滤镜建立阴影效果，使元素内容在页面上产生投影，从而实现立体的效果。其原理就是创建一个偏移量，并定义一个阴影颜色。

DropShadow 滤镜语法格式如下。

```
{filter : DropShadow ( enabled=bEnabled , color=sColor , offx=iOffsetx, offy=iOffsety,
                        positive=bPositive ) }
```

其参数如表 18-4 所示。

表 18-4　DropShadow 滤镜参数及说明

参数	说明
enabled	设置滤镜是否激活
color	指定滤镜产生的阴影颜色
offx	指定阴影水平方向偏移量，默认值为 5px
offy	指定阴影垂直方向偏移量，默认值为 5px
positive	指定阴影透明程度，为布尔值。True（1）表示为任何的非透明像素建立可见的阴影；False（0）表示为透明的像素部分建立透明效果

【例 18.5】（实例文件：ch18\18.5.html）

```
<html>
<head>
    <title>DropShadow 滤镜</title>
</head>
<body>
```

```
        <table width="90%" height="90%">
            <tr>
                <td style="filter: DropShadow(color=gray,offx=10,offy=10,positive=1)">
                    <img src="9.jpg" >
                </td>
            </tr>
            <tr>
                <td style="filter: DropShadow(color=gray,offx=5,offy=5.positive=1);
                        font-size:20pt; color:DarkBlue">
                    这是一个阴影效果
                </td>
            </tr>
        </table>
    </body>
</html>
```

在 IE 9.0 中浏览效果如图 18-5 所示，可以看到图片产生了阴影，但不明显。下方文字产生的阴影效果明显。



图 18-5　阴影效果

18.2.5　FlipH 滤镜

在 CSS 中，可以通过 Filp 滤镜实现 HTML 对象翻转效果，其中 FlipH 滤镜用于水平翻转对象，即将元素对象沿水平方向进行 180° 翻转。FlipH 滤镜可以在 CSS 中直接使用，使用格式如下。

```
{Fliter：FlipH(enabled=bEnabled)}
```

该滤镜中只有一个 enabled 参数，表示是否激活滤镜。

【例 18.6】（实例文件：ch18\18.6.html）

```
<html >
<head>
      <title>FlipH 滤镜</title>
<style>
img{
      height:120px;
      width:200px;
}
</style>
</head>
<body>
            原图片<img src="9.jpg">
            图片水平翻转<img src="9.jpg" style="Filter:FlipH()">

</body>
</html>
```

在 IE 9.0 中浏览效果如图 18-6 所示，可以看到图片进行了左右方向上的翻转。

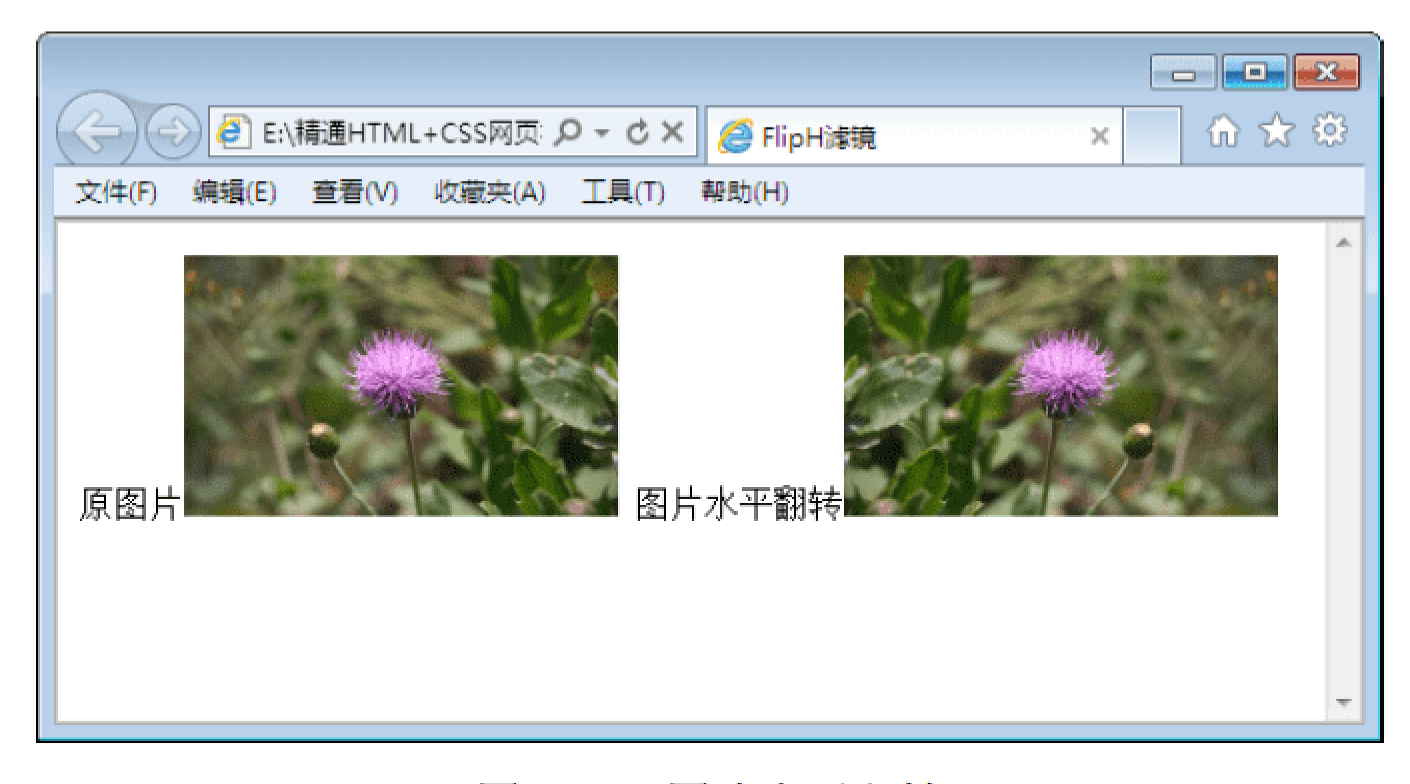


图 18-6　图片水平翻转

18.2.6　Glow 滤镜

文字或物体发光的特性往往能吸引浏览者的注意，Glow 滤镜可以使对象的边缘产生一种柔和的边框或光晕，并可产生如火焰一样的效果。

语法格式如下。

```
{filter : Glow ( enabled=bEnabled , color=sColor , strength=iDistance ) }
```

其中，color 设置边缘光晕颜色，strength 设置晕圈范围，值范围是 1~255，值越大效果越强。

【例 18.7】（实例文件：ch18\18.7.html）

```
<html>
<head>
    <title>filter glow</title>
    这段文字不带有光晕
    <style>
    <!--
    .weny{
        width:100%;
        filter:Glow(color=#9966CC,strength=10)}
    -->
    </style>
</head>
<body>
    <div class="weny">
        <p style="font-family: 1 幼圆; font-size: 40pt; font-weight: bolder; color: #003366">
            这段文字带有光晕
    </div>
</body>
</html>
```

在 IE 9.0 中浏览效果如图 18-7 所示，可以看到文字带有光晕，非常漂亮。当 Glow 滤镜作用于文字时，每个文字边缘都会出现光晕，效果非常强烈。而对于图片，Glow 滤镜只在其边缘加上光晕。

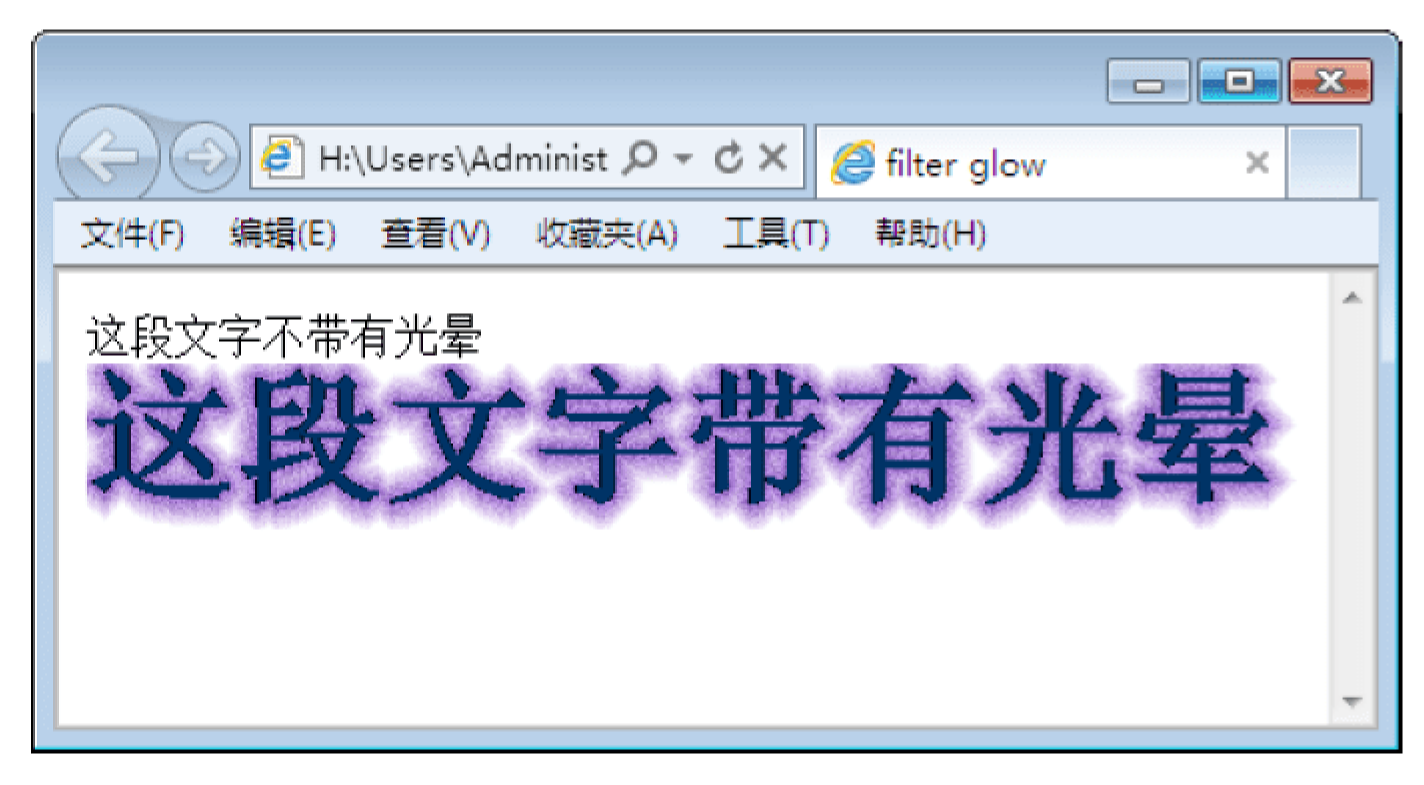


图 18-7　光晕效果

18.2.7　Gray 滤镜

黑白色是一种经典颜色，使用 Gray 滤镜能够轻松地将彩色图片变为黑白图片。

语法格式如下。

```
{filter:Gray(enabled=bEnabled)}
```

该滤镜只有一个参数 enabled，表示是否激活，可以在页面代码中直接使用。

【例 18.8】（实例文件：ch18\18.8.html）

```
<html>
<head>
<title>Gray 滤镜</title>
</head>
<body>
    <img src="9.jpg"    style="width: 50%;height:50%"    />原图
    <img src="9.jpg"   style="width: 50%;height:50%; filter: Gray()"    />    灰度图
</body>
</html>
```

在 IE 9.0 中浏览效果如图 18-8 所示，可以看到下面的图片以黑白色显示。

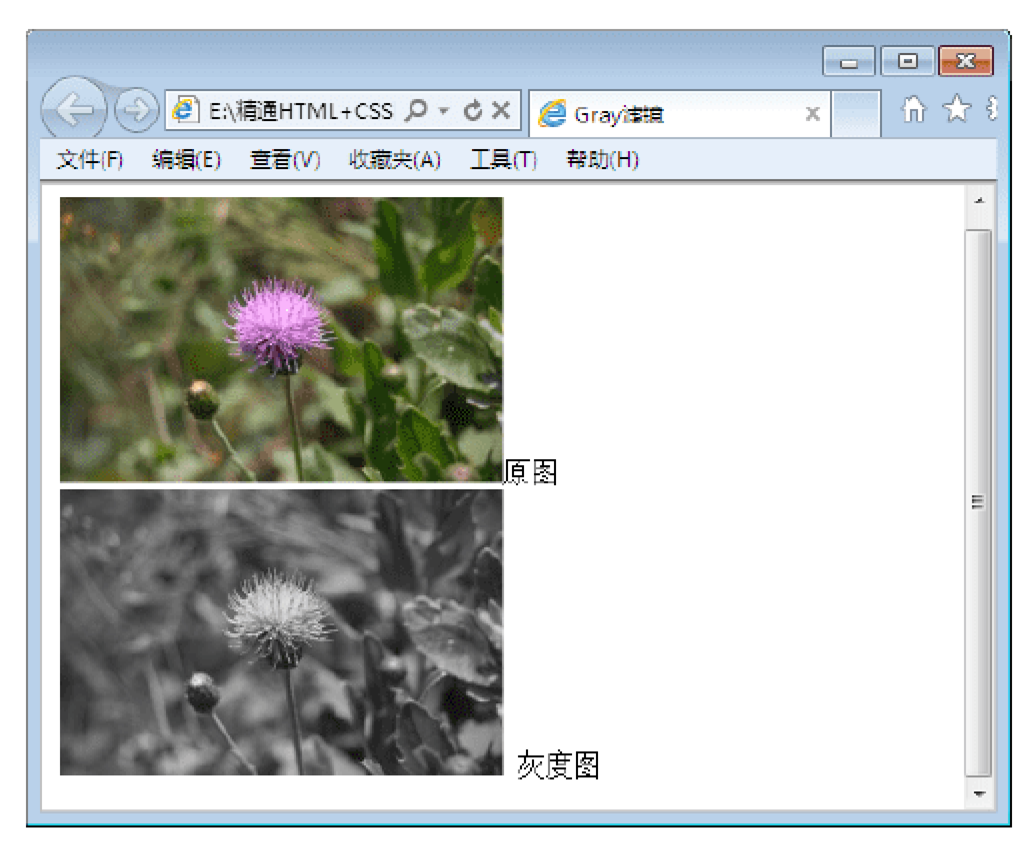


图 18-8　以黑白色显示图片

18.2.8　Invert 滤镜

Invert 滤镜可以把对象的可视化属性全部翻转，包括色彩、饱和度和亮度值，使图片产生一种“底片”或负片的效果。

语法格式如下。

```
{filter:Invert(enabled=bEnabled)}
```

该滤镜只有一个 enabled 参数，用来设置是否激活滤镜。

【例 18.9】（实例文件：ch18\18.9.html）

```
<html>
<head>
<title>Invert 滤镜</title>
</head>
<body>
<img src="9.jpg" />原图
<img src="9.jpg"   style="width:30%; filter: Invert()" />反相图
```

```
</body>
</html>
```

在 IE 9.0 中浏览效果如图 18-9 所示，可以看到下面的图片以相片底片的颜色显示。

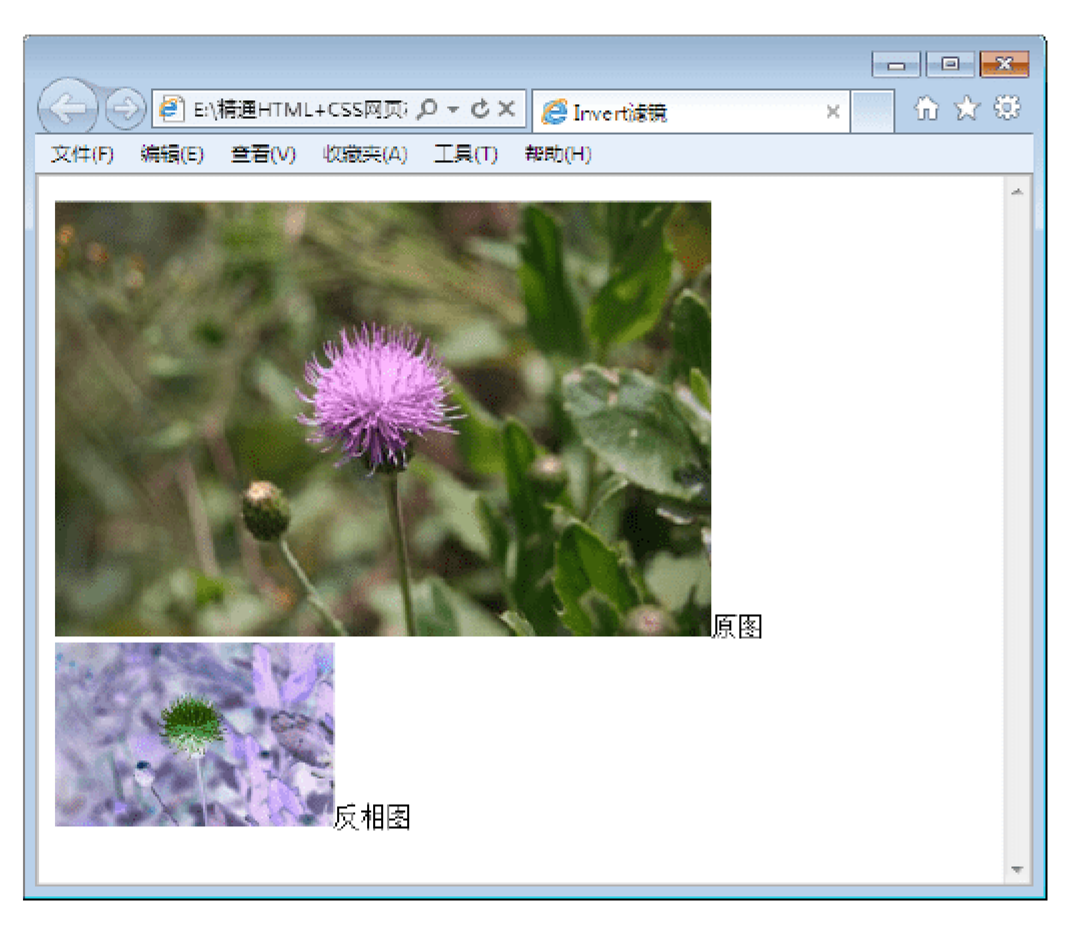


图 18-9　反色效果

18.2.9　FlipV 滤镜

FlipH 滤镜用来实现对象的水平翻转，而 FlipV 滤镜用来实现对象的垂直翻转，其中包括文字和图像。语法格式如下。

```
{Fliter：FlipV(enabled=bEnabled)}
```

该滤镜只有一个 enabled 参数，表示是否激活滤镜。

【例 18.10】（实例文件：ch18\18.10.html）

```
<html>
<head>
<title>FlipV 滤镜</title>
</head>
<body>
        <img src="9.jpg">原图片
        <img src="9.jpg" style="Filter:FlipV()">图片垂直翻转
</body>
</html>
```

在 IE 9.0 中浏览效果如图 18-10 所示，可以看到图片发生了上下翻转。

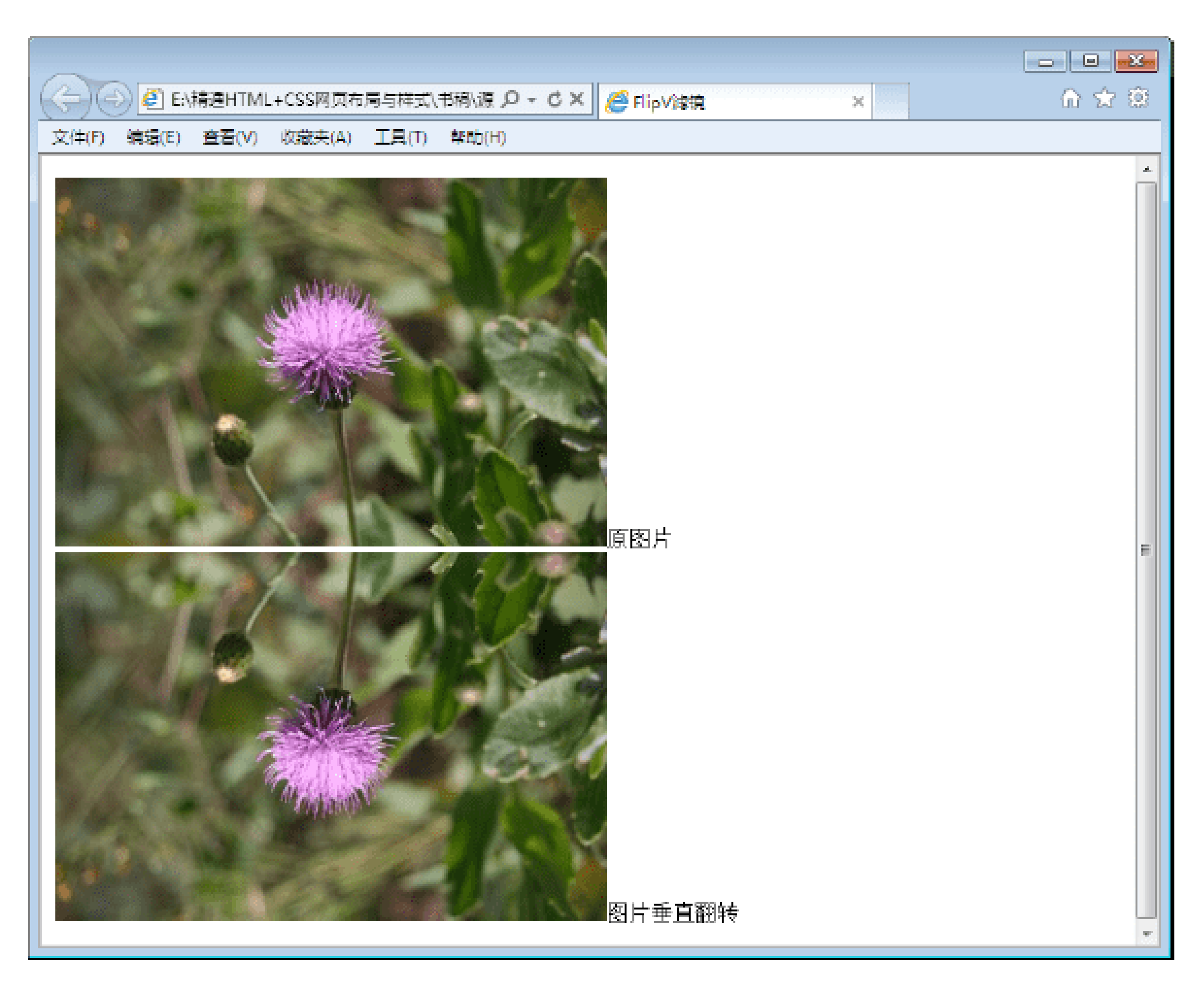


图 18-10　上下翻转

18.2.10　Mask 滤镜

可以通过 mask 滤镜，为网页中的元素对象做出一个矩形遮罩。遮罩，就是使用一个颜色图层将包含文字或图像等对象的区域遮盖，但是文字或图像部分却以背景色显示出来。

Mask 滤镜语法格式如下。

```
{filter:Mask(enabled=bEnabled , color=sColor)}
```

参数 color 用来设置 mask 滤镜作用的颜色。

【例 18.11】（实例文件：ch18\18.11.html）

```
<html>
<head>
<title>Mask 遮罩滤镜</title>
<style>
p {
    width:400;
    filter:mask(color:#FF9900);
    font-size:40pt;
    font-weight:bold;
    color:#00CC99;
}
</style>
</head>
<body>
<p>这里有个遮罩</p>
</body>
</html>
```

在 IE 9.0 中浏览效果如图 18-11 所示，可以看到文字上面有一个遮罩，文字颜色是背景颜色。

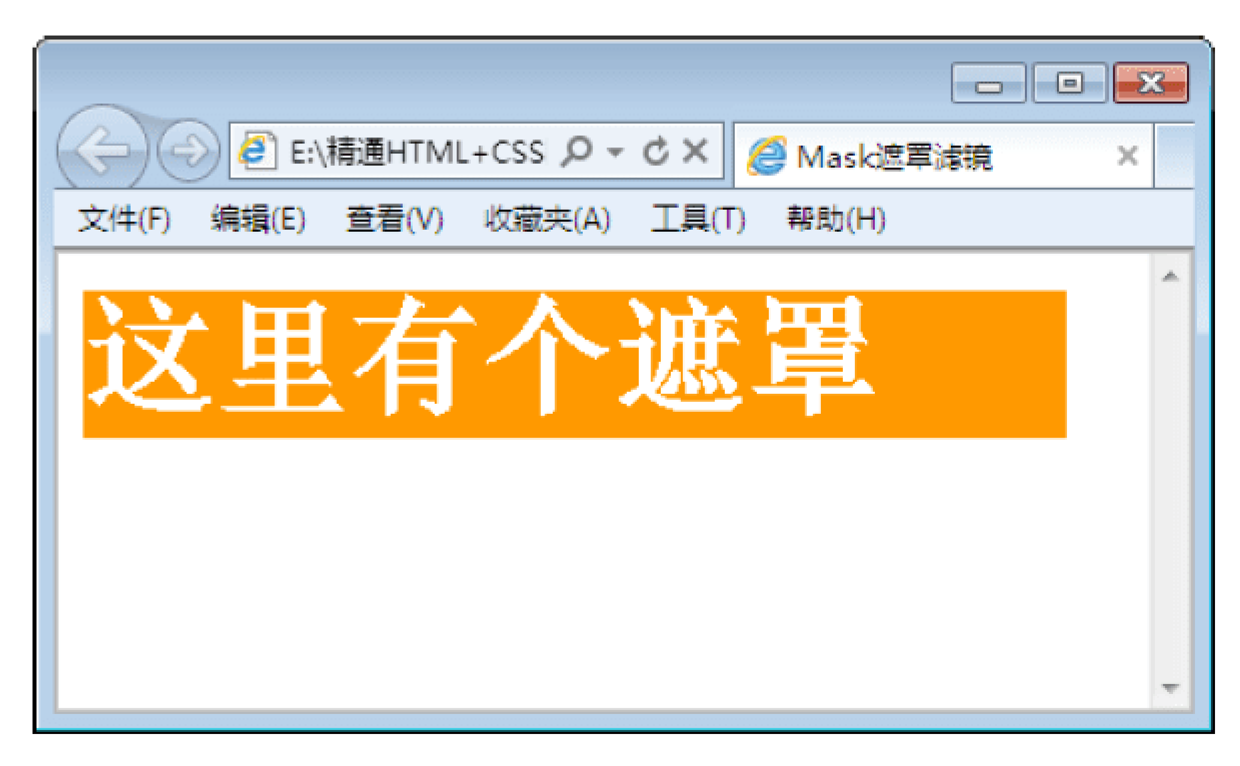


图 18-11　文字遮罩

18.2.11　Shadow 滤镜

可以通过 Shadow 滤镜来给对象添加阴影，其实际效果看起来好像是对象离开了页面，并在页面上显示出该对象阴影。阴影部分的原理是建立一个偏移量，并为其加上颜色。

Shadow 滤镜的语法格式如下。

```
{filter:Shadow(enabled=bEnabled , color=sColor , direction=iOffset, strength=iDistance)}
```

其参数如表 18-5 所示。

表 18-5　Shadow 滤镜参数及说明

参数	说明
enabled	设置滤镜是否激活
color	设置投影的颜色
direction	设置投影的方向，分别有 8 种取值，代表 8 个方向：取值为 0 表示向上方向，45 为右上，90 为右，135 为右下，180 为下方，225 为左下方，270 为左方，315 为左上方。
strength	设置投影向外扩散的距离

【例 18.12】（实例文件：ch18\18.12.html）

```
<html>
<head>
<title>阴影效果</title>
<style>
h1 {
	color:#FF6600;
	width:400;
	filter:shadow(color=blue, offx=15, offy=22, positive=flase);
}
```

```
</style>
</head>
<body>
<h1>开在悬崖上的野百合</h1>
</body>
</html>
```

在 IE 9.0 中浏览效果如图 18-12 所示，可以看到文字带有阴影效果。

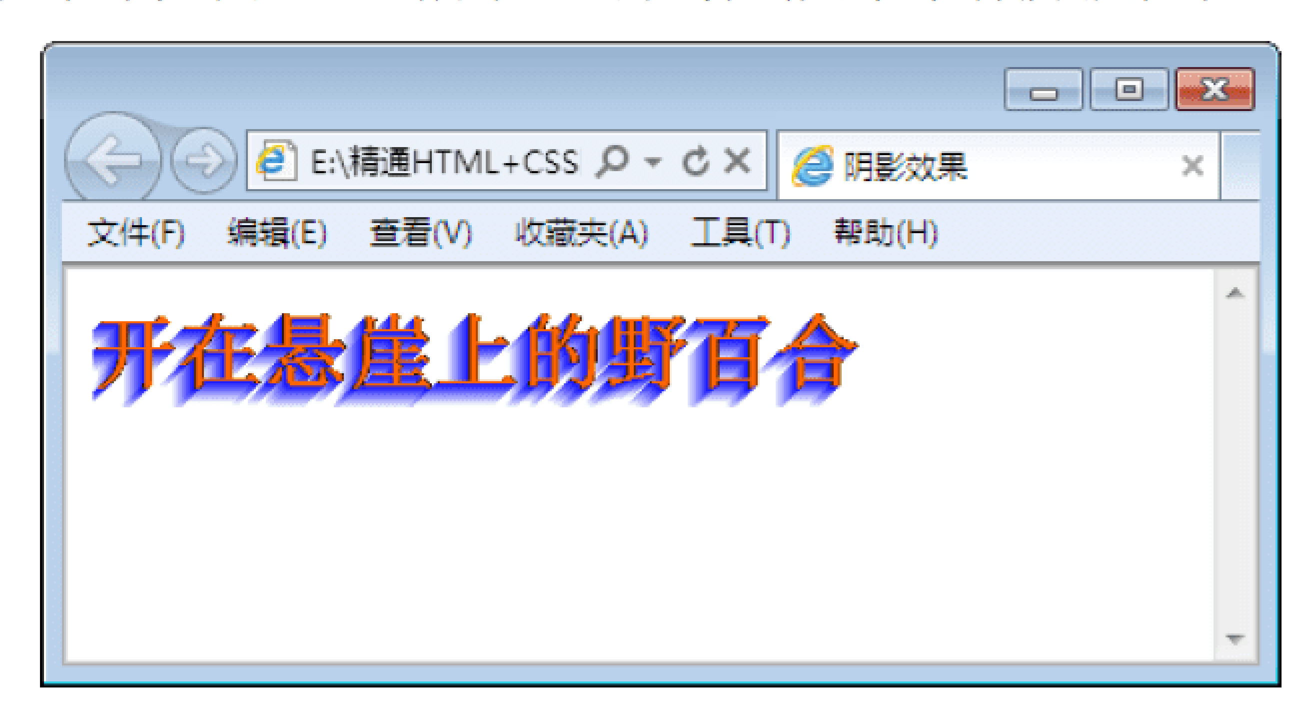


图 18-12　Shadow 滤镜效果

18.2.12　Wave 滤镜

Wave 滤镜可以为对象添加竖直方向上的波浪效果，也可以用来把对象按照竖直的波纹样式打乱。

语法格式如下。

```
{filter:Wave（enabled=bEnabled , add=bAddImage , freq=iWaveCount , lightStrength=iPercentage ,
            phase=iPercentage , strength=iDistance）}
```

各参数说明如表 18-6 所示。

表 18-6　Wave 滤镜参数及说明

参数	说明
enabled	设置滤镜是否激活
add	布尔值，表示是否在原始对象上显示效果。True 表示显示；False 表示不显示
freq	设置生成波纹的频率，也就是设定在对象上产生的完整的波纹的条数
lightStrength	波纹效果的光照强度，取值 0~100
phase	设置正弦波开始的偏移量，取百分比值 0~100，默认值为 0。25 就是 360*25%为 90°，50 则为 180°
strength	波纹的曲折的强度

【例 18.13】（实例文件：ch18\18.13.html）

```
<html>
```

```
<head>
<title>波浪效果</title>
<style>
h1 {
    color:violet;
    text-align:left;
    width:400;
    filter:wave(add=true, freq=5, lightStrength=45, phase=20, strength=3);
}
</style>
</head>
<body>
<h1>一起去看大海</h1>
</body>
</html>
```

在 IE 9.0 中浏览效果如图 18-13 所示，可以看到文字带有波浪效果。

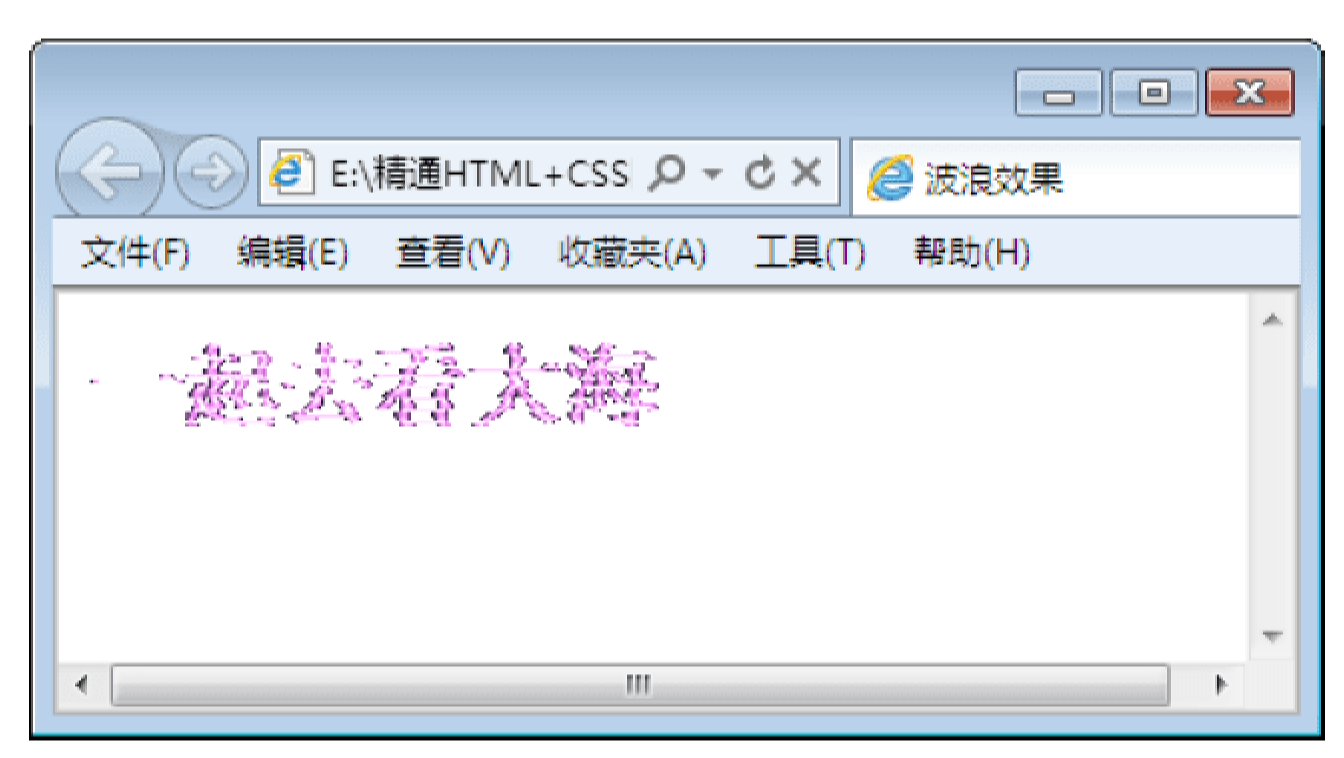


图 18-13　Wave 滤镜效果

18.2.13　X-ray 滤镜

X-ray 中文含义为 X 射线，X-ray 滤镜可以使对象反映出它的轮廓，并把这些轮廓的颜色加亮，使整体看起来有一种 X 光片的效果。

语法格式如下。

```
{filter:Xray(enabled=bEnabled)}
```

该滤镜只有一个 enabled 参数，用于确定是否激活滤镜。

【例 18.14】（实例文件：ch18\18.14.html）

```
<html>
<head>
<title>X 射线</title>
<style>
```

```
.noe {
    filter:xray;
}
</style>
</head>
<body>
<img src="9.jpg" class="noe" />
<img src="9.jpg" />
</body>
</html>
```

在 IE 9.0 中浏览效果如图 18-14 所示，可以看到图片有 X 光效果。

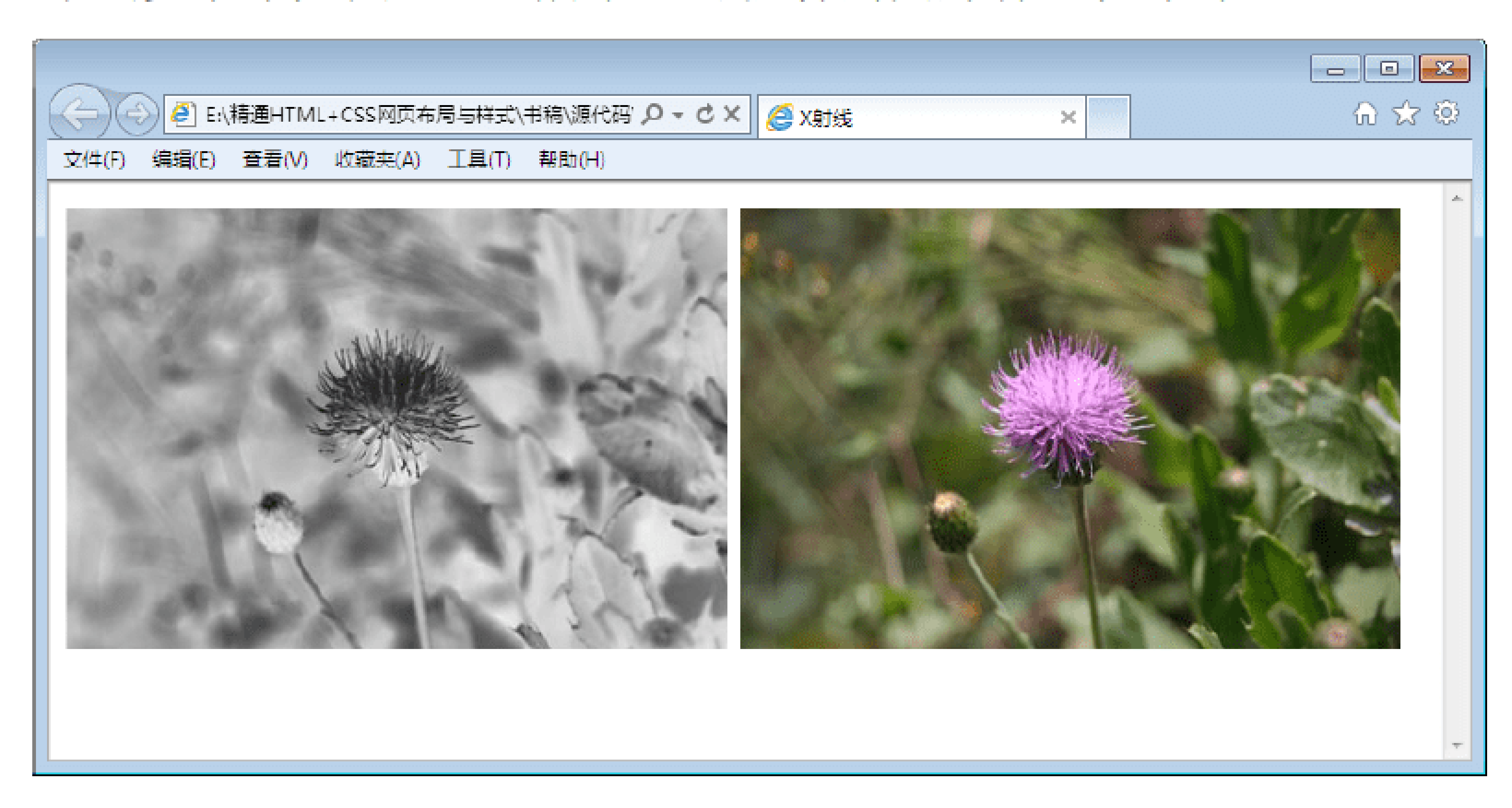


图 18-14　X-ray 滤镜效果

18.3　CSS 的高级滤镜

高级滤镜是指需要配合 JavaScript 脚本语言，产生变换效果的滤镜，包含 BlendTrans、RevealTrans 和 Light 等。

18.3.1　BlendTrans 滤镜

BlendTrans 滤镜是一种高级滤镜，如果要实现滤镜效果，需要结合 JavaScript。该滤镜可以实现 HTML 对象的渐隐渐现效果。

BlendTrans 滤镜语法格式如下。

```
{ filter : BlendTrans ( enabled=bEnabled , duration=fDuration ) }
```

上述代码中，enabled 表示是否激活滤镜；duration 表示整个转换过程所需要的时间，单位为秒。

【例 18.15】（实例文件：ch18\18.15.html）

```
<html >
```

```
<head>
    <title>BlendTrans 滤镜</title>
    <style type="text/css">
    <!--
      .blendtrans { filter:blendTrans(Duration=3)}
    -->
    </style>
</head>
<body onload="playImg()">
  <img src="baimd.jpg" class="blendtrans" id="imgpic"  style="width:300px;height:280px;" />
  <script language="JavaScript">
  <!--
    // 声明数组，数组元素的个数就是图片的个数，然后给数组元素赋值，值为图片路径
    ImgNum=new ImgArray(2);
    ImgNum[0]="baimd.jpg";
    ImgNum[1]="baihua.jpg";
    // 获取数组记录数
    function ImgArray(len)
    {
      this.length=len;
    }
    var i=1;
    //转换过程
    function playImg(){
      if (i==1){
        i=0 ;
      }
      else{
        i++;
      }
      imgpic.filters[0].apply();
      imgpic.src=ImgNum[i];
      imgpic.filters[0].play();
      // 设置演示时间，这里是以毫秒为单位的，4000 则表示延迟 4 秒
      // 滤镜中设置的转换时间值，这样当转换结束后还要停留一段时间
      timeout=setTimeout('playImg()',4000);
    }
    -->
    </script>
</body>
</html>
```

上述代码中，对 HTML 元素 img 应用了 BlendTrans 滤镜，然后使用 Javascript 脚本语言来定义转换过程。对于 JavaScript 代码，要声明用来存储图片数组，并指定图片所在路径。然后，再获取数组长度，用于在转换过程中循环读取图片数量。接着定义转换过程 playImg，该过程实现了两幅图片之间的淡入淡出并进行转换的过程，apply 方法用于捕获对象内容的初始显示，为转换做必要的准备。timeout 指定了转换的延迟时间，再加上滤镜中设置的转换时间，则图片在转换之间将停留，以方便清楚地浏览图片。最后，在主体元素 body 中插入 onload 事件，加载转换过程。

在 IE 9.0 中浏览效果如图 18-15 所示，可以看到一张图片慢慢消失，另一张图片慢慢出现，两张图片不断循环往复，从而实现渐变效果。

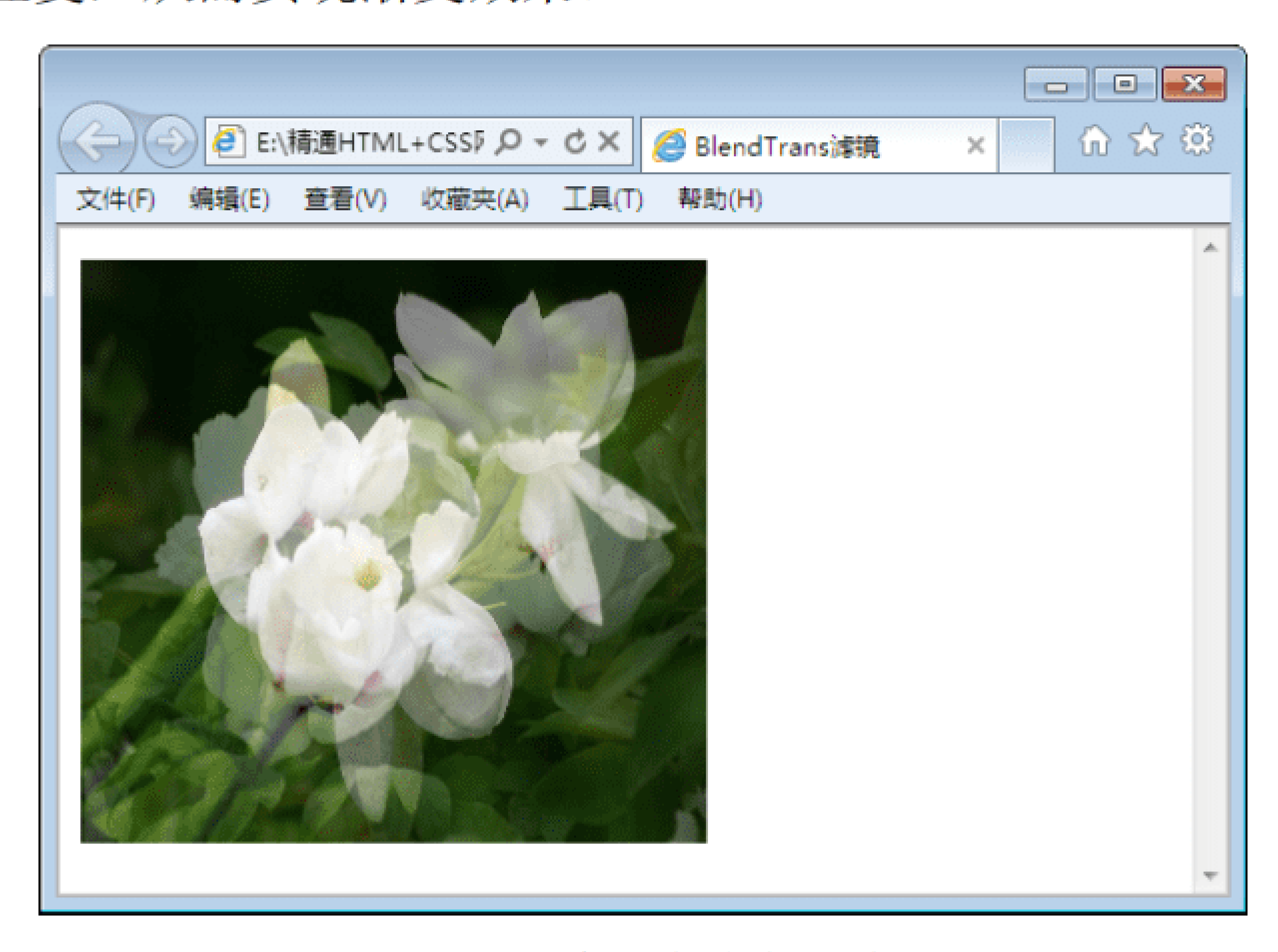


图 18-15　实现渐隐渐现效果

18.3.2　Light 滤镜

Light 滤镜是一个高级滤镜，需要结合 JavaScript 使用。该滤镜用来产生类似于光照灯的效果，并可调节亮度以及颜色。

语法格式如下。

```
{filter:Light(enabled=bEnabled)}
```

对于已定义的 Light 滤镜属性，可以调用它的方法来设置或改变属性。这些方法如表 18-7 所示。

表 18-7　Light 滤镜方法及说明

方法	说明
AddAmbIE	加入包围的光源
AddCone	加入锥形光源
AddPoint	加入点光源
Changcolor	改变光的颜色

（续表）

方法	说明
Changstrength	改变光源的强度
Clear	清除所有的光源
MoveLight	移动光源

【例 18.16】（实例文件：ch18\18.16.html）

```
<html>
<head>
<title>light 滤镜效果</title>
</head>
<body>
    <table>
        <tr>
            <td style="color:blue; font-weight:bolder">
                随鼠标变化的动态光源效果
            </td>
        </tr>
        <tr>
            <td id="light" style="filter: light(); width: 200px">
                <img src="feng.jpg">
            </td>
        </tr>
    </table>
    <script language="Javascript">
    <!--
      var g_numlights=0;
      // 调用设置光源函数
      window.onload=setlights;
      // 获得鼠标句柄
      light.onmousemove=mousehandler;
      //建立光源的集合
      function setlights(){
          light.filters[0].clear();
          light.filters[0].addcone(0,0,5,100,100,255,255,0,60,30);
      }
      // 捕捉鼠标的位置来移动光线焦点
      function mousehandler(){
          x=(window.event.x-80);
          y=(window.event.y-80);
          light.filters[0].movelight(0,x,y,5,1);
```

```
            }
        -->
        </script>
    </body>
    </html>
```

在 IE 9.0 中浏览效果如图 18-16 所示，可以看到一幅实现光照效果的图片，随着鼠标的移动，灯光照射的方向也不相同，类似于镭射灯的效果。

实现光照效果时，JavaScript 脚本语言起主要作用。首先要创建光源并指定光源位置。setlights 函数中 filters[0]表示设置的光源滤镜，调用 clear 方法表示在每次页面加载时先清除所有的光源，然后再使用 addcone 方法创建锥形光源。如果需要在图片上添加多束光源，则可以重复使用 addcone 方法，但注意要使用不同的参数，否则光源处于同一位置，就无法产生效果了。函数 mousehandler 用来实现光束随着鼠标移动的效果。

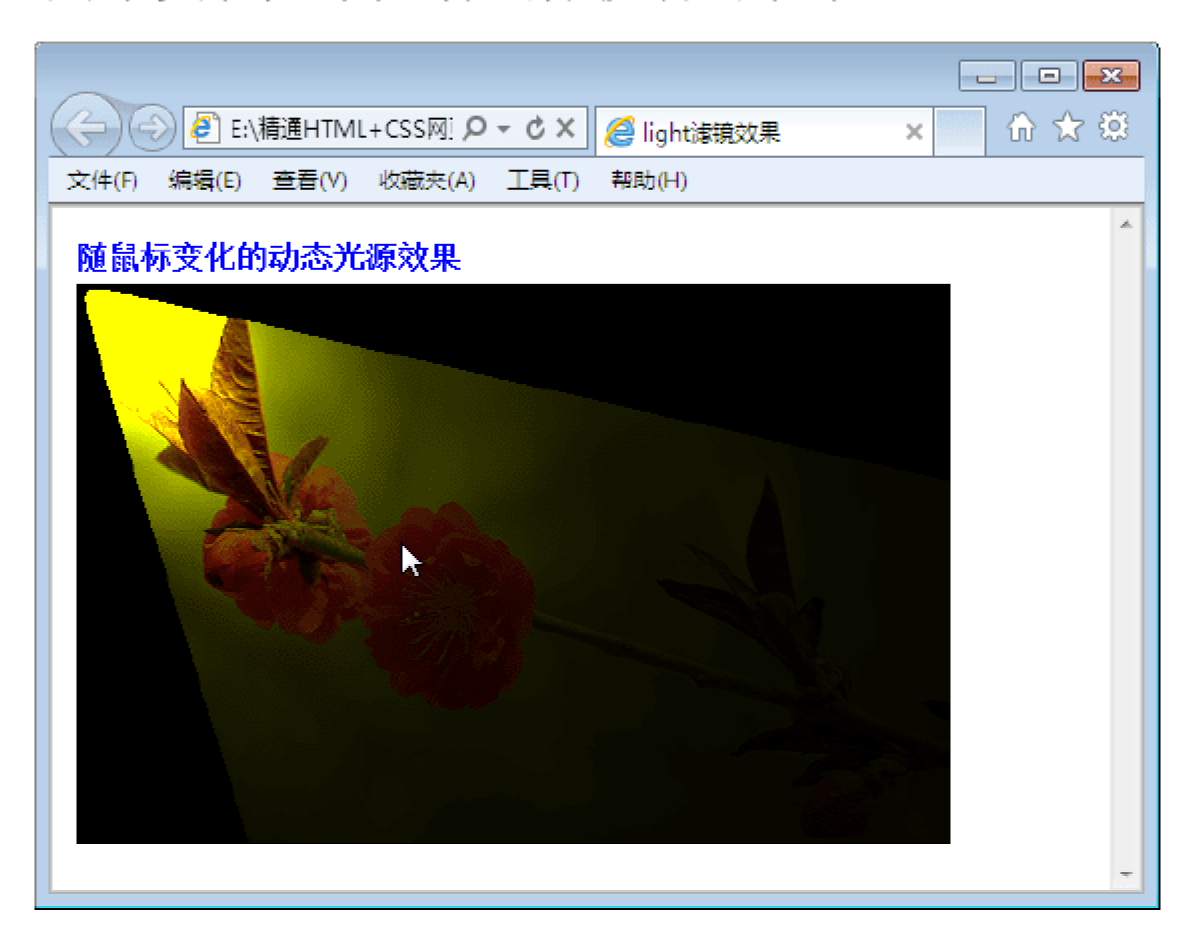


图 18-16　Light 滤镜效果

18.3.3　RevealTrans 滤镜

RevealTrans 滤镜能够实现图像之间的切换效果。进行图像切换时，能产生 32 种动态效果，例如，溶解、水平（垂直）展幕、百叶窗等，可以随机选取其中一种效果进行切换。

RevealTrans 滤镜语法格式如下。

```
filter : RevealTrans ( enabled=bEnabled , duration=fDuration , transition=iTransitionType )
```

其中，enabled 表示是否激活滤镜；duration 用于设置切换停留时间；transition 用于指定转换方式，即指定要使用的动态效果，参数取值是 0~23。transition 参数值及效果如表 18-8 所示。

表 18-8　RevealTrans 滤镜动态效果及其对应的 transition 参数值

动态效果	参数值	动态效果	参数值
矩形从大至小	0	随机溶解	12
矩形从小至大	1	从上下向中间展开	13

（续表）

动态效果	参数值	动态效果	参数值
圆形从大至小	2	从中间向上下展开	14
圆形从小至大	3	从两边向中间展开	15
向上推开	4	从中间向两边展开	16
向下推开	5	从右上向左下展开	17
向右推开	6	从右下向左上展开	18
向左推开	7	从左上向右下展开	19
垂直形百叶窗	8	从左下向右上展开	20
水平形百叶窗	9	随机水平细纹	21
水平棋盘	10	随机垂直细纹	22
垂直棋盘	11	随机选取一种效果	23

但是，如果只设置了 transition 参数来实现切换过程的话，是不会有任何效果的，因为动态效果的实现还必须依靠脚本语言 JavaScript 调用相应的方法。

【例 18.17】（实例文件：ch18\18.17.html）

```
<html >
<head>
      <title> RevealTrans 滤镜</title>
<style type="text/css">
   .revealtrans { filter:revealTrans(Transition=10,Duration=3)}
</style>
</head>
<body onload="playImg()">
   <img id="imgpic" class="revealtrans" src="9.jpg"/>
   <script language="JavaScript">
   <!--
      // 声明数组，数组元素的个数就是图片的个数，然后给数组元素赋值，值为图片路径
      ImgNum=new ImgArray(2);
      ImgNum[0]="baimd.jpg";
      ImgNum[1]="baihua.jpg";
      // 获取数组记录数
      function ImgArray(len)
      {
         this.length=len;
      }
      var i=1;
      //转换过程
      function playImg(){
         if (i==1){
```

```
            i=0 ;
        }
        else{
            i++;
        }
        imgpic.filters[0].apply();
        imgpic.src=ImgNum[i];
        imgpic.filters[0].play();
        // 设置演示时间，这里是以毫秒为单位的，4000 则表示延迟 4 秒
        // 滤镜中设置的转换时间值，这样当转换结束后还停留一段时间
        timeout=setTimeout('playImg()',4000);
    }
    -->
    </script>
</body>
</html>
```

对于图片切换，实现代码与 BlendTrans 滤镜的转换代码相似，这里不再介绍。

在 IE 9.0 中浏览效果如图 18-17 所示，可以看到图片以百叶窗的方式打开另一幅图片，如此循环往复。

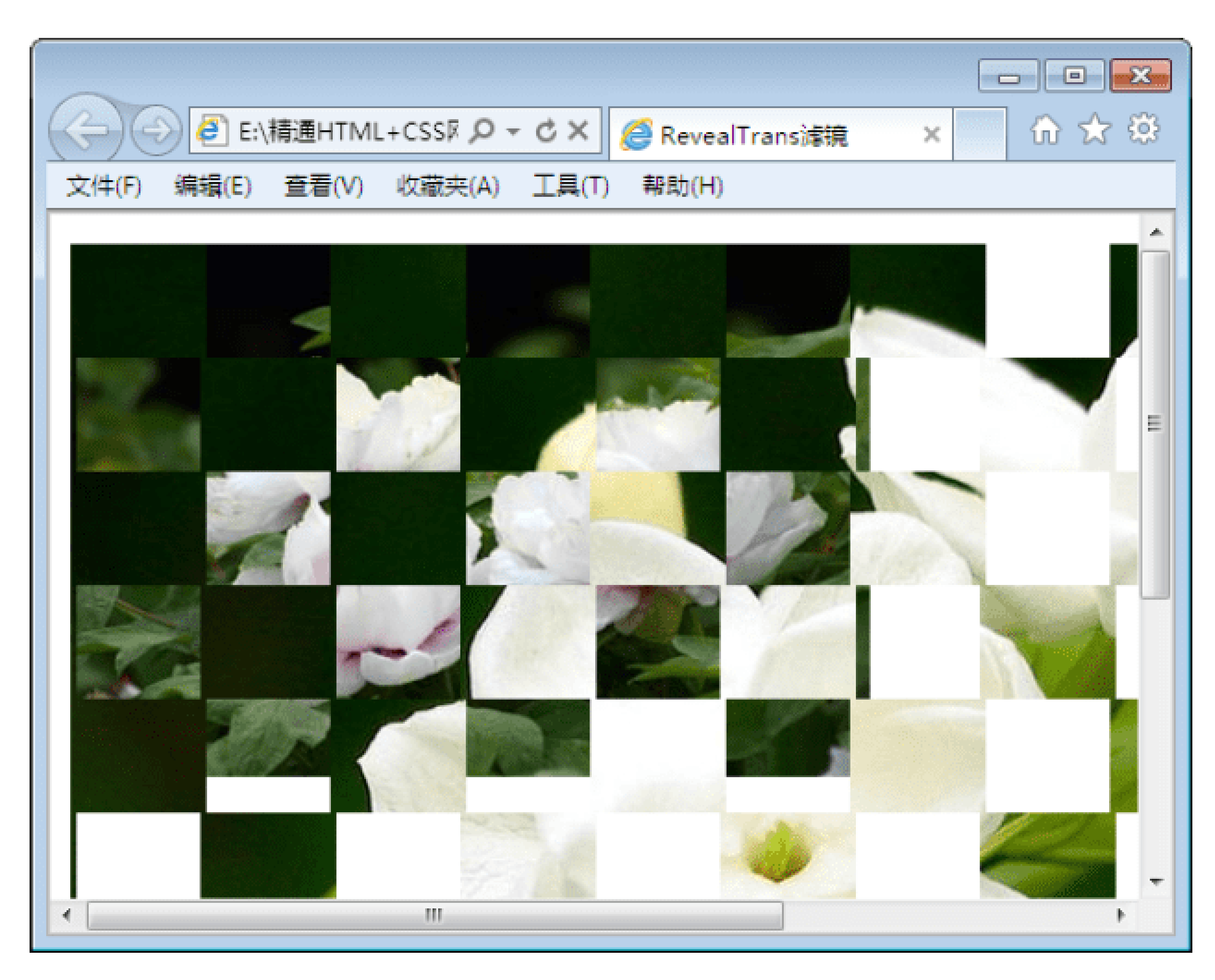


图 18-17　图片切换效果

18.4　专家解惑

1．如果要对一个 HTML 对象使用滤镜效果，应如何使用？

在使用滤镜时，若使用多个滤镜，则每个滤镜之间用空格分隔开；一个滤镜中的若干个参数用逗号分隔；filter 属性和其他样式属性同时使用时以分号分隔。

2．滤镜效果是 IE 9.0 浏览器特有的 CSS 特效，那么在 Firefox 中能不能实现呢？

滤镜效果虽然是 IE 9.0 浏览器特有效果，但使用 Firefox 浏览器的属性也可以实现相同效果。例如，对于 IE 9.0 的阴影效果，在 Firefox 网页设计中，可以先在文字下面再叠加一层浅色的同样的文字，然后做 2 个像素的错位，制造阴影的假象。

第 19 章

CSS定位与DIV布局

在网页设计中，能否很好地定位网页中的每个元素，是网页整体布局的关键。一个布局混乱、元素定位不准的页面，是每个浏览者都不喜欢的。因此把每个元素都精确定位到合理位置，才是构建美观大方页面的前提。

19.1　定位方式

在 CSS 中，定位可以将一个元素精确地放在页面上用户所指定的位置，而布局是将整个页面的元素内容整洁且完美地摆放。定位的实现是布局成功的前提。如果清晰地掌握了定位原理，那么就能够创建多种高级而精确的布局，并会让网页的实现更加完美。

19.1.1　定位属性

在网页设计中，定位（Positioning）原理简单，就是用户精确地定义 HTML 元素框所在页面中的位置，可以是页面绝对位置，也可以是对于其上级元素、另一个元素或浏览器窗口的相对位置。

可以认为每个元素都包含在一个矩形框内，称为元素框。而元素内容与元素框共同形成了元素块。所谓定位，就是定位元素块的位置和大小。要实现 CSS 定位，需要依赖定位属性才能够完成。

表 19-1 列出了 CSS 中全部有关的定位属性。

表 19-1　定位属性及其含义

定位属性	含义
position	定义位置
left	指定元素横向距左部距离
right	指定元素横向距右部距离
top	指定元素纵向距顶部距离
bottom	指定元素纵向距底部距离
z-index	设置元素的层叠顺序
width	设置元素框宽度

（续表）

定位属性	含义
Height	设置元素框高度
overflow	内容溢出控制
clip	剪切

表 19-1 中前 6 个属性是实际的定位属性，后面的 4 个有关属性是用来设置元素框，或对元素框中的内容进行控制的属性。其中，position 属性是最主要的定位属性。它既可以定义元素框的绝对位置，又可以定义相对位置，而 left、right、top 和 bottom 只有在 position 属性中使用才会发挥作用。

19.1.2 position 属性

网页中各种元素需要有自己合理的位置，从而搭建整个页面的结构。在 CSS 中，可以通过 position 属性，对页面中的元素进行定位。

其语法格式如下所示：

```
position : static | absolute | fixed | relative
```

其参数含义如表 19-2 所示。

表 19-2 position 属性的参数及说明

参数	说明
static	元素定位的默认值，无特殊定位，对象遵循 HTML 定位规则，不能通过 z-index 进行层次分级
relative	相对定位，对象不可重叠，可以通过 left、right、bottom 和 top 等属性在正常文档中偏移位置，可以通过 z-index 进行层次分级
absolute	生成绝对定位的元素，相对于 static 定位以外的第一个父元素进行定位。元素的位置通过 left、top、right 以及 bottom 属性设置
fixed	fixed 生成绝对定位的元素，相对于浏览器窗口进行定位。元素的位置通过 left、top、right 以及 bottom 属性进行设置

1. 绝对定位（absolute）

绝对定位是参照浏览器的左上角，配合 top、left、bottom 和 right 进行定位的，如果没有设置上述 4 个值，则默认以父级的坐标原点为原始点。绝对定位可以通过上、下、左、右来设置元素，使之处在任何一个位置。

在父层 position 属性为默认值时，上、下、左、右的坐标原点以 body 的坐标原点为起始位置。绝对定位的语法格式如下所示：

```
position:absolute
```

只要将上面代码加入到样式中，使用样式的元素就可以以绝对定位的方式显示了。

【例 19.1】（实例文件：ch19\19.1.html）

```
<html>
<head>
<title>定位属性</title>
</head>
<body>
  <div style="background-color: Black; width:200px; height:200px">
    <h2 style=" position:absolute; left:80px; top:80px; width:110px; height:50px;
        background-color:Red;">这是绝对定位</h2>
  </div>
</body>
</html>
```

在 IE 9.0 中浏览效果如图 19-1 示，可以看到红色元素框以浏览器左上角为原点，坐标位置为（80px，80px），宽度为 110 像素，高度为 50 像素。

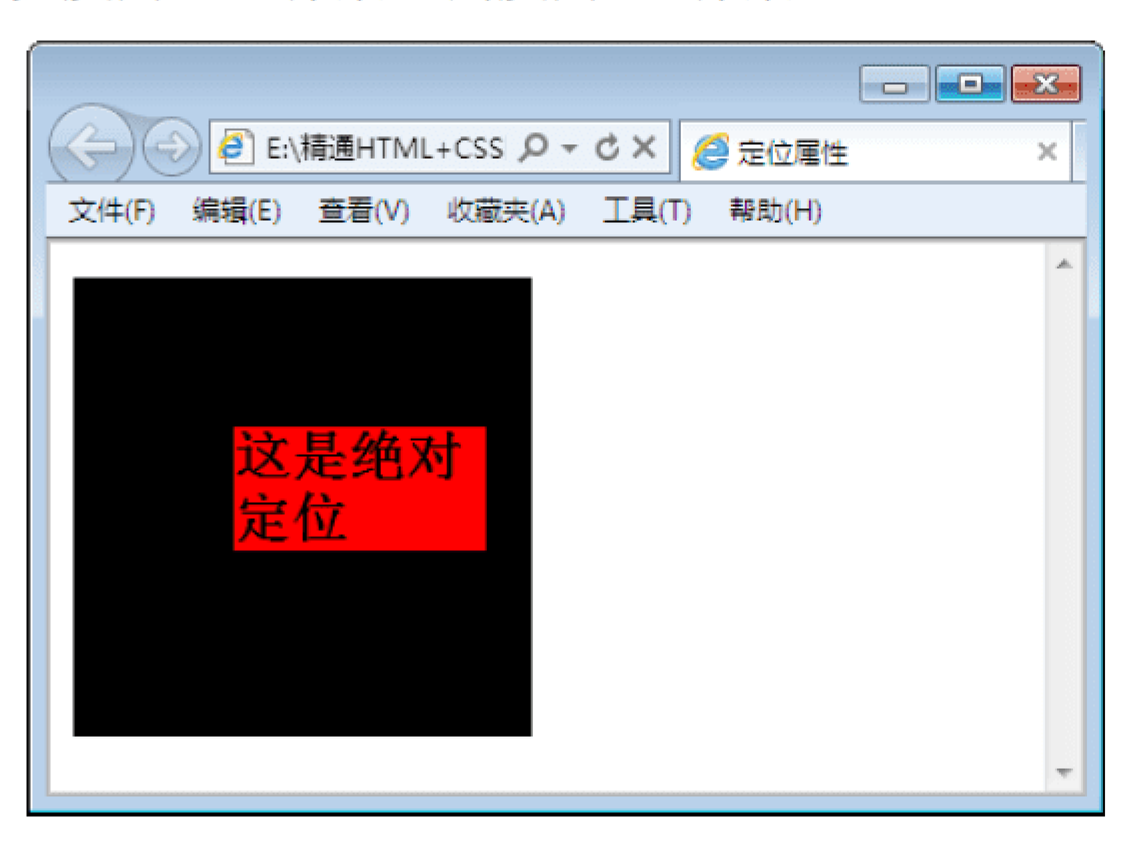


图 19-1　绝对定位

优秀的页面设计能够适用于各种屏幕分辨率，并且要能够保证正常的显示。要解决这个问题，在定位时最好使用相对定位。

2．相对定位（relative）

如果对一个元素进行相对定位，首先它将出现在它所在的原始位置上。然后通过设置垂直或水平位置，让这个元素“相对于”它的原始起点进行移动。另外，进行相对定位时，无论是否进行移动，元素仍然占据原来的空间。因此，移动元素会导致它覆盖其他框。

绝对定位与相对定位的区别在于：绝对定位的坐标原点为上级元素的原点，与上级元素有关；相对定位的坐标原点为本身偏移前的原点，与上级元素无关。

相对定位的语法格式如下所示：

```
position:relative
```

【例 19.2】（实例文件：ch19\19.2.html）

```
<html>
<head>
<style type="text/css">
h2.pos_left
{
position:relative;
left:-20px
}
h2.pos_right
{
position:relative;
left:20px
}
</style>
</head>
<body>
<h2>这是位于正常位置的标题</h2>
<h2 class="pos_left">这个标题相对于其正常位置向左移动</h2>
<h2 class="pos_right">这个标题相对于其正常位置向右移动</h2>
<p>相对定位会按照元素的原始位置对该元素进行移动。</p>
<p>样式 "left:-20px" 从元素的原始左侧位置减去 20 像素。</p>
<p>样式 "left:20px" 向元素的原始左侧位置增加 20 像素。</p>
</body>
</html>
```

在 IE 9.0 中浏览效果如图 19-2 所示，可以看到页面显中示了 3 个标题，最上面标题正常显示，下面两个标题分别以正常标题为原点，分别向左和向右移动了 20 像素。

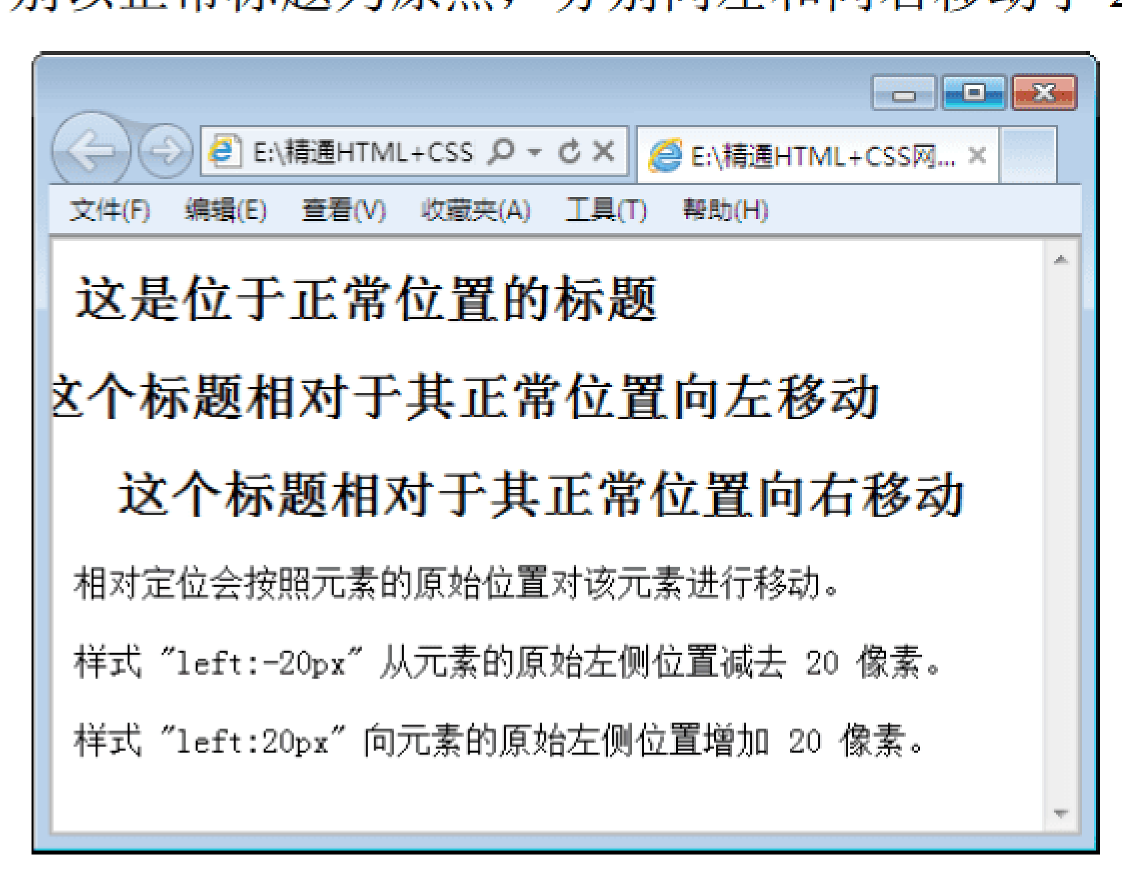


图 19-2　相对定位

3．固定定位（fixed）

固定定位和绝对定位比较相似，它是绝对定位的一种特殊形式，固定定位的容器不会随着

滚动条的拖动而变换位置。在视线中，固定定位的容器位置是不会改变的。固定定位可以把一些特殊效果固定在浏览器的视线位置。

固定定位的参照位置不是上级元素块而是浏览器窗口。所以可以使用固定定位来设定类似传统框架样式布局，以及广告框架或导航框架等。使用固定定位的元素可以脱离页面，无论页面如何滚动，始终处在页面的同一位置上。

固定定位语法格式如下所示。

```
position:fixed
```

【例 19.3】（实例文件：ch19\19.3.html）

```
<html>
<head>
<title>CSS 固定定位</title>
<style type="text/css">...
* {
padding:0;
margin:0;
}
#fixedLayer {
width:100px;
line-height:50px;
background: #FC6;
border:1px solid #F90;
position:fixed;
left:10px;
top:10px;
}
</style>
</head>
<body>
<div id="fixedLayer">固定不动</div>
<p>我动了</p>
<p>我动了</p>
<p>我动了</p>
<p>我动了</p>
<p>我动了</p>
<p>我动了</p>
<p>我动了</p>
<p>我动了</p>
<p>我动了</p>
<p>我动了</p>
<p>我动了</p>
```

```
<p>我动了</p>
</body>
</html>
```

在 IE 9.0 中浏览效果如图 19-3 所示，可以看到拉动滚动条时，无论页面内容怎么变化，其黄色框“固定不动”始终处在页面左上角顶部。

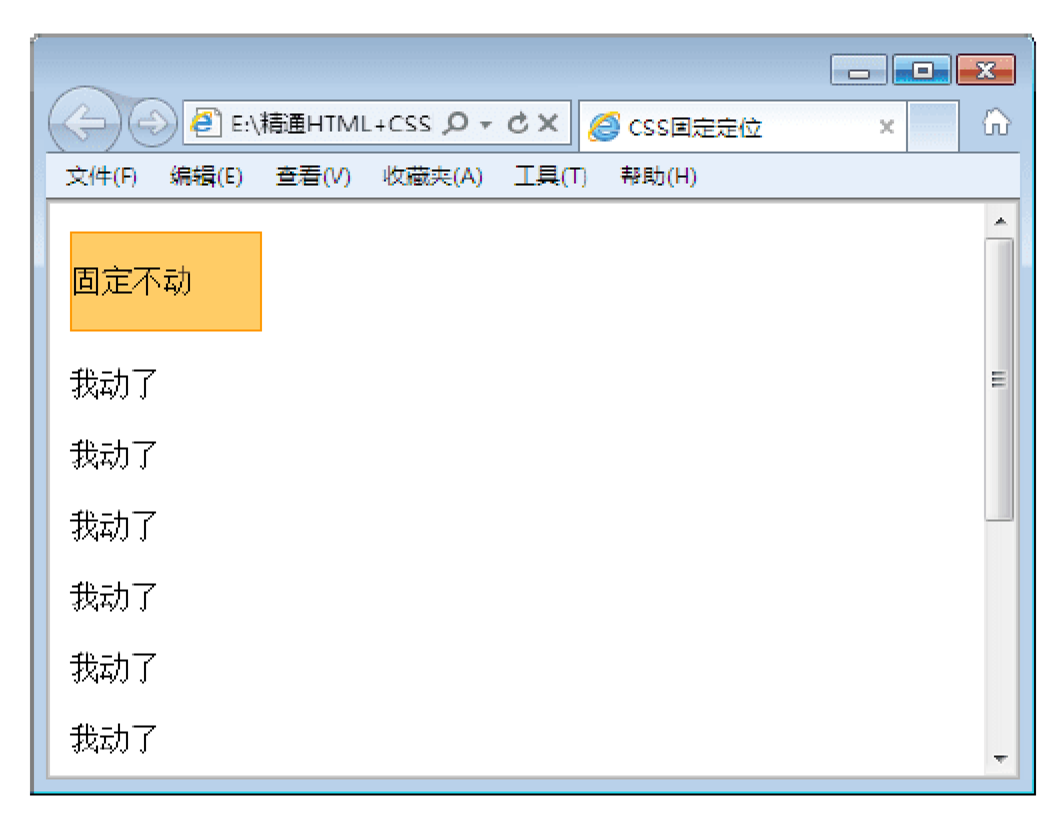


图 19-3　固定定位

19.1.3　层叠顺序

对 HTML 元素进行定位时，可以从其高度、宽度和深度 3 个方面入手，设置高度时使用 height、宽度使用 width、深度使用 z-index。z-index 用来设置元素层叠的次序，其方法是为每个元素指定一个数字，数字较大的元素将叠加在数字较小的元素之上。

z-index 语法格式如下所示：

```
z-index : auto | number
```

其参数值 auto 表示遵循父对象的定位，number 是一个无单位的整数值，可以为负值。如果两个决定定位元素的 z-index 属性具有相同的 number 值，则依据该元素在 HTML 文档中声明的顺序进行层叠。如果绝对定位的元素没有指定 z-index 属性，则此属性的 number 值为正数的元素会叠加在该元素之上，而 number 值为负数的对象在该元素之下。如果将参数设置为 null，可以消除此属性。该属性只作用于 position 的属性值为 relative 或 absolute 的对象，不能作用在窗口组件上。

【例 19.4】（实例文件：ch19\19.4.html）

```
<html>
<head>
<title>Z-index 使用</title>
<style>
#big {
    width:800px;
    height:220px;
```

```
        padding:6px;
        background-color:#999999;
        position:relative;
}
#Div1 {
        width:160px;
        height:80px;
        background-color:#FFD700;
        padding:6px;
        position:absolute;
        left:9px;
        top:9px;
        z-index:6;
}
#Div2 {
        width:120px;
        height:80px;
        background-color:thistle;
        padding:6px;
        position:absolute;
        left:280px;
        top:90px;
        z-index:4;
}
#Div3 {
        width:140px;
        height:80px;
        background-color:lightskyblue;
        padding:6px;
        position:absolute;
        left:150px;
        top:25px;
        z-index:5;
}
</style>
</head>
<body>
<div id="big">
   <div id="Div1"><br>
      z-index 值是 6 ; </div>
   <div id="Div2"><br>
       z-index 值是 4 ; </div>
```

```
    <div id="Div3"><br>
      z-index 值是  5 ; </div>
</div>
</body>
</html>
```

在 IE 9.0 中浏览效果如图 19-4 所示，可以看到网页中显示了 3 个层，3 个层中数值大小不同，并从大到小分别在其他层上显示。

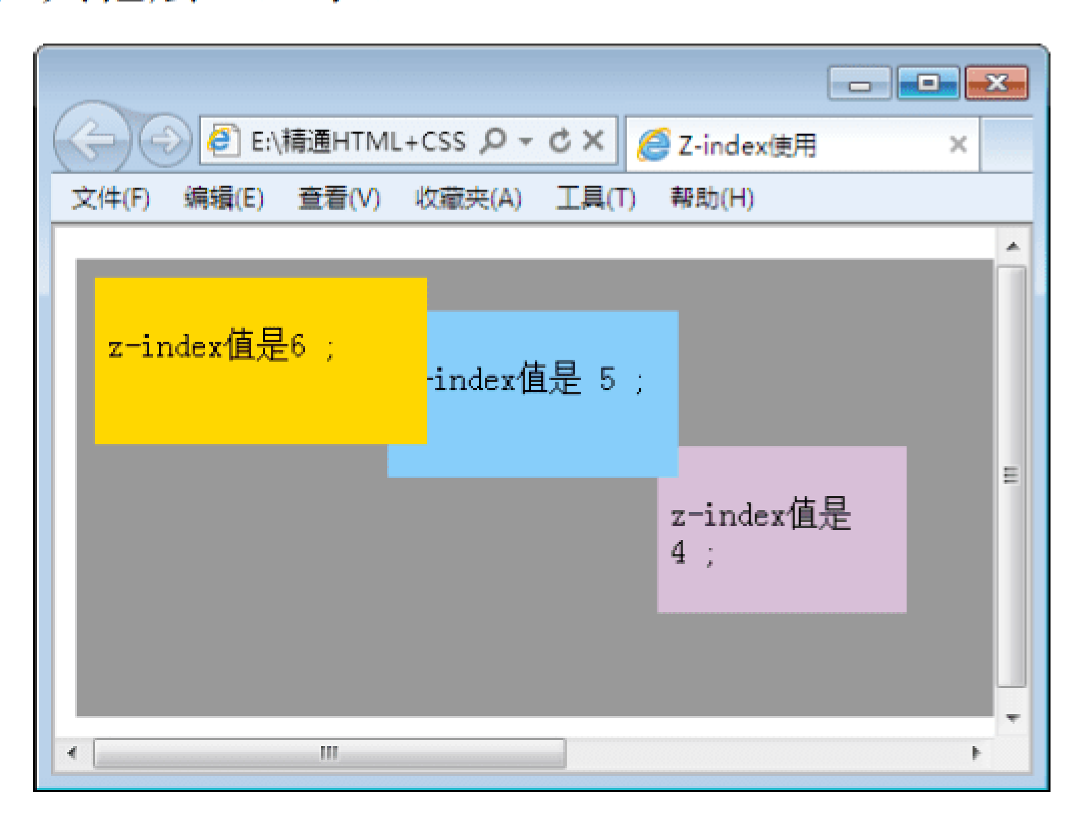


图 19-4　层叠顺序效果显示

19.1.4　边偏移属性

边偏移属性包含 left、right、top 和 bottom 属性。所谓边偏移属性，就是用来描述元素块与包含元素块最近的边线之间的偏移量的属性。其中 left 描述元素块最左边与包含其的边框最左边的边线的距离，如果 left 属性值为正，则会偏右移，如果为负，则会使它更向左移，甚至移出边线。其他依次类推。

left、right、top 和 bottom 4 个属性取值非常相似，这里以 left 为例介绍。left 语法格式如下所示：

```
left : auto | length
```

上面参数值中 auto 表示系统自动取值，length 表示由浮点数字和单位标识符组成的长度值或百分数。直接设定数值用来设置元素的绝对位置，一旦该位置确定，那么该元素将始终处于页面中的该位置。使用百分比设置元素位置，是相对于其上级元素的位置而设定的。如果取值为 auto，则在定位中允许元素刚好有显示其内容所需的宽度及高度，而不必再指明宽度及高度的值。

【例 19.5】（实例文件：ch19\19.5.html）

```
<html>
<head>
<title>定位属性</title>
</head>
```

```
<body>
  <div style="background-color: Black; width:200px; height:200px">
    <p style=" position:relative; left:50%; right:0; top:50%; bottom:0; width:100px; height:100px;
        background-color:Red;">hello
    </p>
  </div>
</body>
</html>
```

在 IE 9.0 中浏览效果如图 19-5 所示，可以看到红色正方形框在指定位置显示，其底侧和右侧分别和大的矩形框对应。

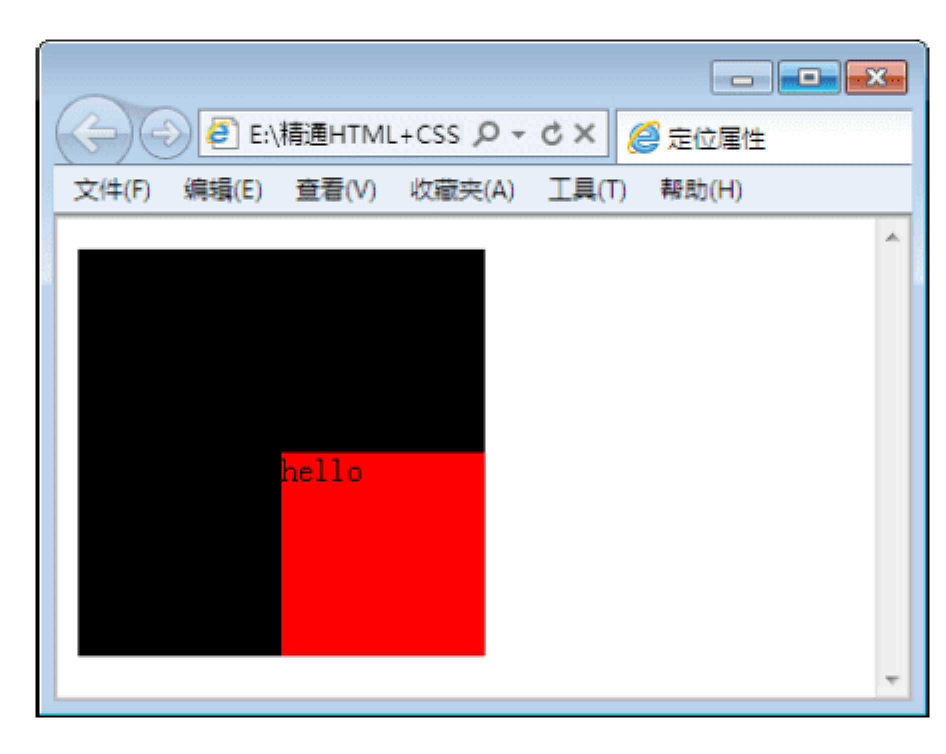


图 19-5　边偏移效果显示

19.2　浮动定位

除了使用 position 进行定位外，还可以使用 float 定位。Float（浮动）定位只能在水平方向上定位，而不能在垂直方向上定位。float 属性表示浮动属性，它用来改变元素块的显示方式。

float 语法格式如下所示：

```
float : none | left |right
```

属性值如表 19-3 所示。

表 19-3　float 属性值及说明

属性值	说明
none	元素不浮动
left	浮动在左面
right	浮动在右面

实际上，使用 float 可以实现两列布局，也就是让一个元素在左面浮动，一个元素在右面浮动，并能控制好这两个元素的宽度。

【例 19.6】（实例文件：ch19\19.6.html）

```
<html>
<head>
<title>float 定位</title>
<style>
* {
	padding:0px;
	margin:0px;
}
.big {
	width:600px;
	 height:100px;
	margin:0 auto 0 auto;
	 border:#332533 1px solid;

}
.one {
	width:300px;
	 height:20px;
	float:left;
	border:#996600 1px solid;
}
.two {
	width:290px;
	 height:20px;
	float:right;
	margin-left:5px;
	display:inline;
	border:#FF3300 1px solid;
}
</style>
</head>
<body>
<div class="big">
   <div class="one">
   <p>非诚勿扰</p>
   </div>
   <div class="two">
   <p>中国达人秀</p>
   </div>
</div>
```

```
</body>
</html>
```

在 IE 9.0 中浏览效果如图 19-6 所示，可以看到显示了一个大矩形框，大矩形框中存在两个小的矩形框，并列显示。

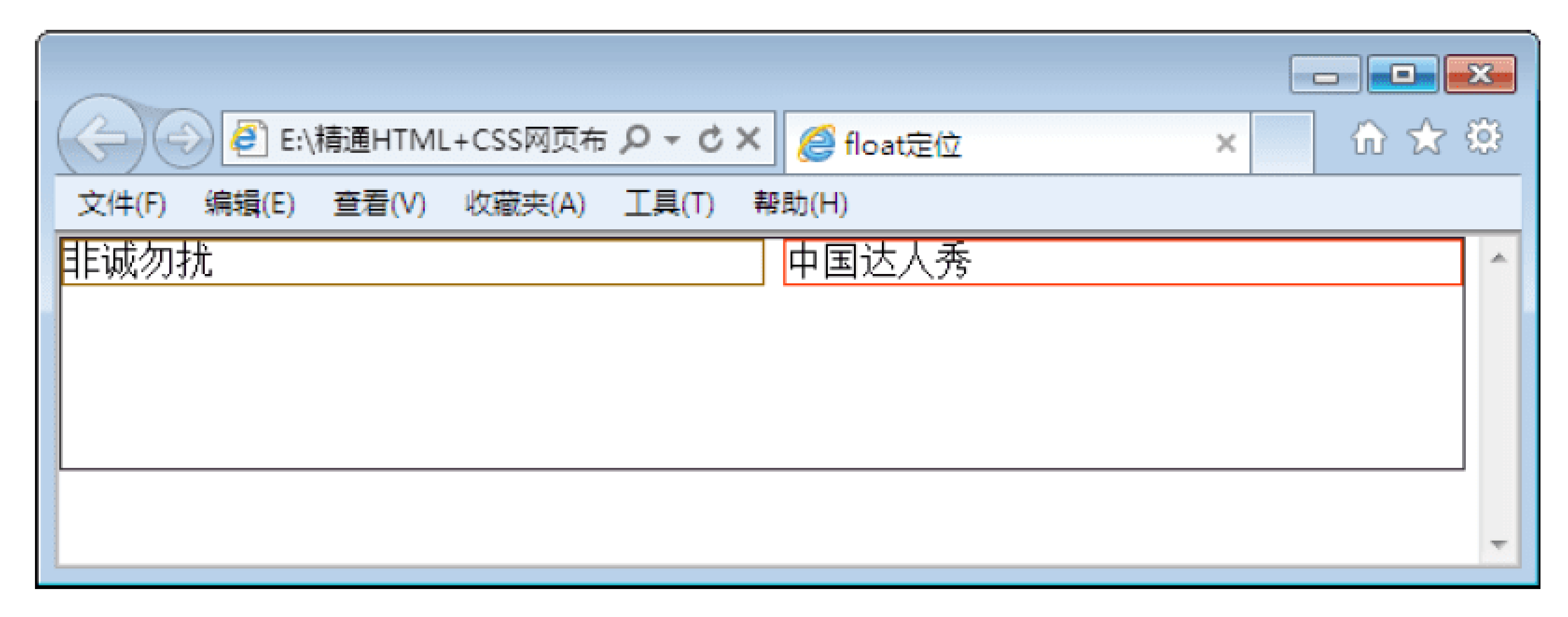


图 19-6 float 浮动布局

使用 float 属性不但可以改变元素的显示位置，同时会对相邻内容造成影响。定义了 float 属性的元素会覆盖其他元素，而被覆盖的区域将处于不可见状态。使用该属性能够实现内容环绕图片的效果。

如果不想让 float 下面的其他元素浮动环绕在该元素周围，可以使用 CSS 的 clear 属性，消除这些浮动元素。

clear 语法格式如下所示：

```
clear : none | left |right | both
```

其中，none 表示允许两边都可以有浮动对象，both 表示不允许有浮动对象，left 表示不允许左边有浮动对象，right 表示不允许右边有浮动对象。使用 float 以后，在必要的时候就需要通过 clear 语句清除 float 带来的影响，以免出现“其他 DIV 跟着浮动”的效果。

19.3 溢出定位

如果元素框被指定了大小，而元素的内容不适合该大小，例如元素内容较多，元素框显示不下，此时则可以使用溢出属性 overflow 来控制这种情况。

overflow 语法格式如下所示：

```
overflow : visible | auto | hidden | scroll
```

各属性值及其说明如表 19-4 所示。

表 19-4 overflow 属性值及说明

属性值	说明
visible	若内容溢出，则溢出内容可见
hidden	若内容溢出，则溢出内容隐藏
scroll	保持元素框大小，在框内应用滚动条以显示内容
auto	等同于 scroll，它表示在需要时应用滚动条

overflow 属性适用于以下情况。

- 元素有负边界。
- 元素框宽于上级元素内容区，换行不被允许。
- 元素框宽于上级元素区域宽度。
- 元素框高于上级元素区域高度。
- 元素定义了绝对定位。

【例 19.7】（实例文件：ch19\19.7.html）

```
<html>
<head>
      <title>overflow 属性</title>
      <style >
          div{
                position:absolute;
                color:#445633;
                height:200px;
                width: 30%;
                float:left;
                margin: 0px;
                padding: 0px;
                border-right: 2px dotted #cccccc;
                border-bottom: 2px solid #cccccc;
                padding-right: 10px;
                overflow:auto;
          }
      </style>
</head>
<body >
    <div>
          <p>综艺节目排名</p><p>1  非诚勿扰</p><p>2  康熙来</p>
          <p>3   快乐大本营</p><p>4   娱乐大风暴</p><p>5 天天向上</p><p>6  爱情连连看</p>
          <p>7  锵锵三人行</p><p>8  我们约会吧</p>
    </div>
```

```
</body>
</html>
```

在 IE 9.0 中浏览效果如图 19-7 所示，可以看到在一个元素框中显示了多个元素，拉动显示的滚动条可以查看全部元素。如果 overflow 设置的值为 hidden，则会隐藏多余元素。

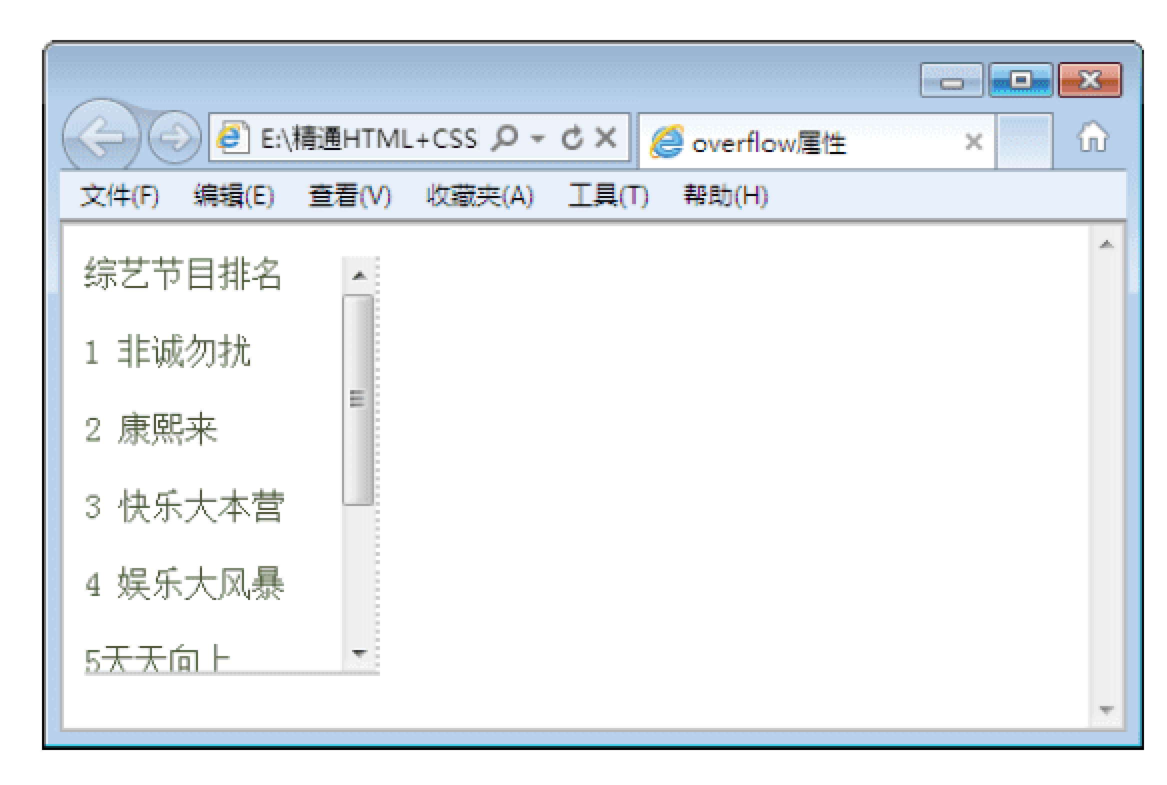


图 19-7　溢出定位

19.4　隐藏定位

visibility 属性指定是否显示一个元素生成的元素框。这意味着元素仍占据其本来的空间，不过可以完全不可见，即设定元素的可见性。

visibility 语法格式如下所示：

```
visibility : inherit | visible | collapse | hidden
```

其属性值如表 19-5 所示。

表 19-5　visibility 属性值及说明

属性值	说明
visible	元素可见
hidden	元素隐藏
collapse	主要用来隐藏表格的行或列。隐藏的行或列能够被其他内容使用。对于表格外的其他对象，其作用等同于 hidden

如果元素 visibility 属性的属性值设定为 hidden，表现为元素隐藏，即不可见。但是，元素不可见并不等同于元素不存在，它仍旧会占有部分页面位置，影响页面的布局。换句话说，元素仍然处于页面中自己的位置上，只是无法看到它而已。

【例 19.8】（实例文件：ch19\19.8.html）

```
<html>
<head>
  <title>float 属性</title>
```

```
    <style type="text/css">
      .div{
        padding:5px;
      }
      .pic{
        float:left;
        padding:20px;
          visibility:visible;
      }
      h1{
          font-weight:bold;
          text-align:center
      }
    </style>
  </head>
  <body>
    <h1>插花</h1>
    <div class="div">
      <div class="pic">
        <img src="08.jpg"   width=150px height=100px />
      </div>
        <p>插花就是把花插在瓶、盘、盆等容器里，而不是栽在这些容器中。所插的花材，或枝、或花、或叶，均不带根，只是植物体上的一部分，并且不是随便乱插的，而是根据一定的构思来选材，遵循一定的创作法则，插成一个优美的形体（造型），借此表达一种主题，传递一种感情和情趣，使人看后赏心悦目，获得精神上的美感和愉快。
        </p>
        <p>
         在我国插花的历史源远流长，发展至今已为人们日常生活所不可缺少。一件成功的插花作品，并不是一定要选用名贵的花材、高价的花器。一般看来并不起眼的绿叶，一个花蕾，甚至路边的野花野草，常见的水果、蔬菜，都能插出一件令人赏心悦目的优秀作品来。使观赏者在心灵上产生共鸣的是创作者唯一的目的，如果不能产生共鸣那么这件作品也就失去了观赏价值。具体地说，即插花作品在视觉上首先要立即引起一种感观和情感上的自然反应，如果未能立刻产生反应，那么摆在眼前的这些花材将无法吸引观者的目光。在插花作品中引起观赏者情感产生反应的要素有三点：一是创意（或称立意)、指的是表达什么主题，应选什么花材；二是构思(或称构图)，指的是这些花材怎样巧妙配置造型，在作品中充分展现出各自的美：三是插器，指的是与创意相配合的插花器皿。三者有机配合，作品便会给人以美的享受。
    </p>
    </div>
  </body>
</html>
```

在 IE 9.0 中浏览效果如图 19-8 所示，可以看到图片在页面左边显示，并被文本信息所环绕。此时 visibility 属性为 visible，表示图片可见。



图 19-8　隐藏定位显示

19.5　块和行内元素

在网页设计中，根据需要可以将段落设置成一个块显示，并带有边框，即类似于 DIV 层的效果。也可以把多个 HTML 元素放在同一行显示。本节将介绍这两种实现方式。

19.5.1　块元素

在 CSS 中可以通过 display 属性，控制元素显示方式。

display 语法格式如下所示：

```
display : block | none | inline | compact | marker | inline-table | list-item | run-in | table |table-caption | table-cell | table-column | table-column-group | table-footer-group | table-header-group | table-row | table-row-group
```

属性值及说明如表 19-6 所示。

表 19-6　display 属性值及说明

属性值	说明
block	以块元素方式显示
inline	以内联元素方式显示
none	元素隐藏
list-item	以列表方式显示
compact	CSS2 分配对象为块对象或基于内容之上的内联对象
marker	CSS2 指定内容在容器对象之前或之后。要使用此参数，对象必须和:after 及:before 伪元素一起使用
inline-table	CSS2 将表格显示为无前后换行的内联对象或内联容器
list-item	CSS1 将块对象指定为列表项目。并可以添加可选项目标志
run-in	CSS2 分配对象为块对象或基于内容之上的内联对象

（续表）

属性值	说明
table	CSS2 将对象作为块元素级的表格显示
table-caption	CSS2 将对象作为表格标题显示
table-cell	CSS2 将对象作为表格单元格显示
table-column	CSS2 将对象作为表格列显示
table-column-group	CSS2 将对象作为表格列组显示
table-header-group	CSS2 将对象作为表格标题组显示
table-footer-group	CSS2 将对象作为表格脚注组显示
table-row	CSS2 将对象作为表格行显示
table-row-group	CSS2 将对象作为表格行组显示

display 属性的默认值为 block，即元素的默认显示方式是以块元素方式显示。常用的段落 p、标题 h1、表单 form、列表 ul 和列表选项 li 都可以定义成块元素。块元素的行高、顶部和底部都是可控制的。如果不设置宽度的话，块元素会默认为整个容器的 100%，如果设定了值，则显示大小由值决定。

【例 19.9】（实例文件：ch19\19.9.html）

```
<title>块元素</title>
<style>
    .big{
        width:800px;
        height:105px;
        background-image:url(07.jpg);
        }
    a{
        font-size:12px;
        display:block;
        width:100px;
        height:20px;
        line-height:20px;
        background-color:#F4FAFB;
        text-align:center;
        text-decoration:none;
        border-bottom:1px dotted #6666FF;
        color:black;
        }
    a:hover{
        font-size:13px;
        display:block;
        width:100px;
```

```
            height:20px;
            line-height:20px;
            text-align:center;
            text-decoration:none;
            color:green;
            }
</style>
</head>
<body>
    <div class="big">
    <p>
    <a href="#">管理应用</a><a href="#">财务管理</a><a href="#">在线管理</a>
    <a href="#">客户关系管理</a><a href="#">一体化管理</a>
   </p>
   </div>
</body>
```

在 IE 9.0 中浏览效果如图 19-9 所示，可以看到左边显示了一个导航栏，右边显示了一个图片。其导航栏就以块元素形式显示。

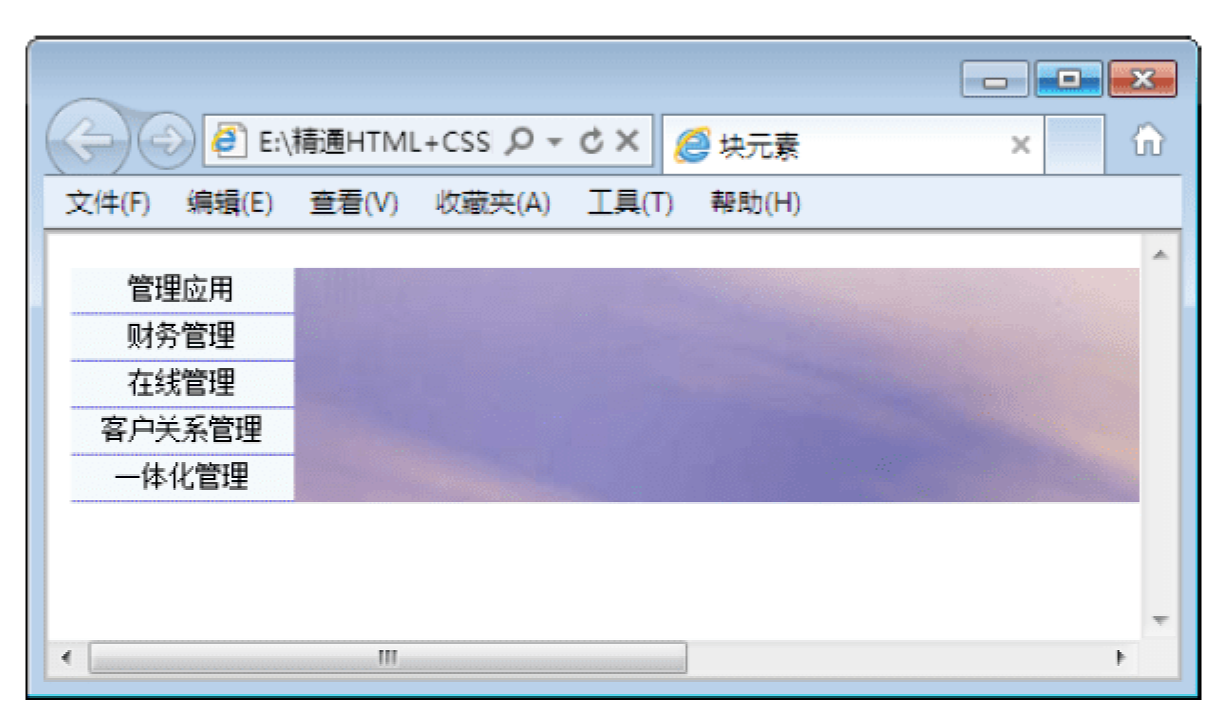


图 19-9　块元素显示

19.5.2　行内元素

当 display 的值被设定为 inline 时，可以把元素设置为行内元素，并在浏览器中同一行显示。inline 元素决定和其他 HTML 元素在同一行上，其行高、顶部和底部边距可以改变，而宽度是不可以改变的。

【例 19.10】（实例文件：ch19\19.10.html）

```
<html>
<head>
<title>行内元素</title>
<style type="text/css">
.hang {
```

```
        display:inline;
}
</style>
</head>
<body>
<div>
<a href="#" class="hang">这是 a 标签</a>
  <span class="hang">这是 span 标签</span>
<strong class="hang">这是 strong 标签</strong>
<img   class="hang" src=6.jpg/>
</div>
</body>
</html>
```

在 IE 9.0 中浏览效果如图 19-10 所示，可以看到页面中显示的 4 个 HTML 元素，都在同一行显示，包括超级链接、文本信息和图片。

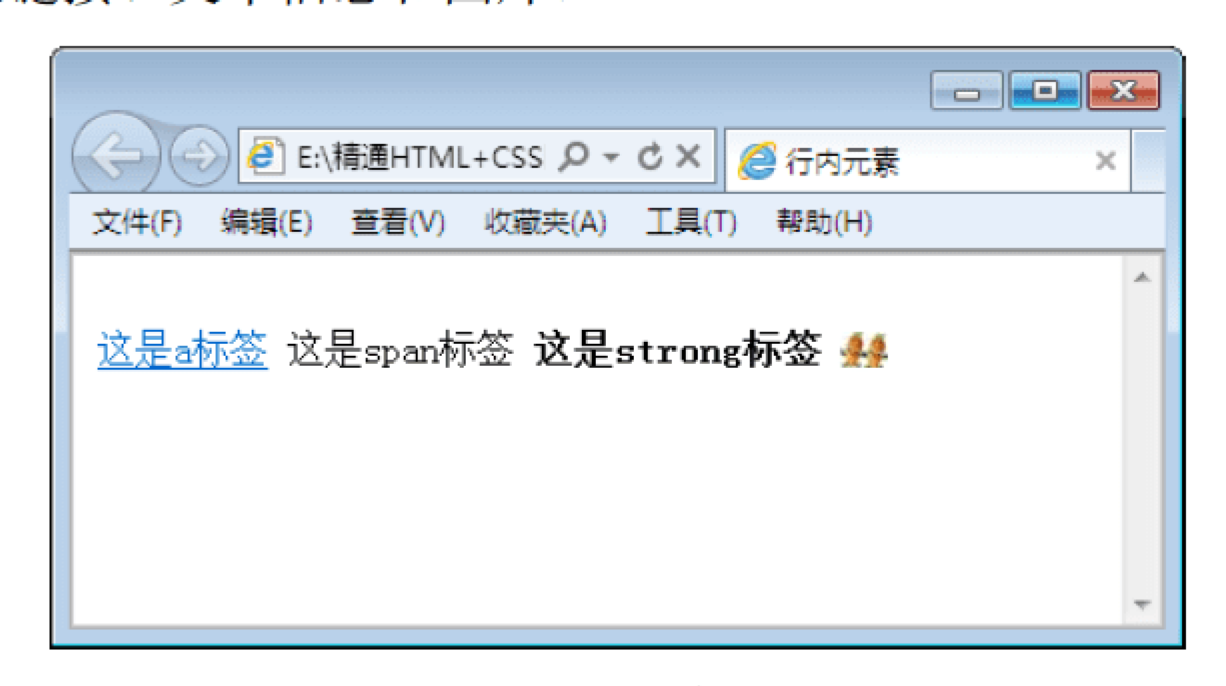


图 19-10 行内显示

19.6 综合实例——左右布局的新闻页面

一个美观大方的页面，必然是一个布局合理的页面。左右布局是网页中比较常见的一种方式，即根据信息种类不同，将信息分别在当前页面左右侧显示。本实例将利用前面学习的知识，创建一个左右布局的新闻页面。

具体步骤如下所示。

01 分析需求。

首先需要将整个页面分为左右两个模块，左模块放置一类信息，右模块放置一类信息。可以设定其宽度和高度。实例完成后，效果如图 19-11 所示。

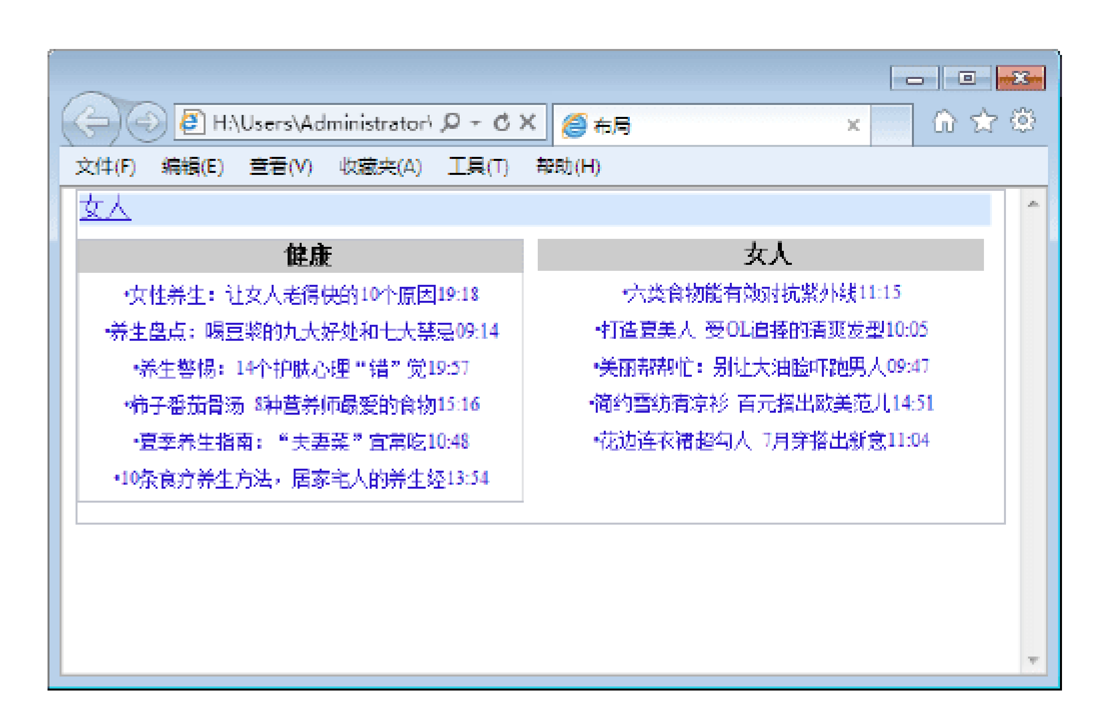


图 19-11　左右显示信息

02 创建 HTML 页面，实现基本列表。

创建 HTML 页面，同时用 DIV 在页面中划分左边 DIV 层和右边 DIV 层两个区域，并且将信息放入到相应的 DIV 层中，注意 DIV 层内引用 CSS 样式名称。

```
<html>
<head>
<title>布局</title>
</head>
<body>
<center>
<div class="big">
  <p class=pp>女人</p>
  <div class="left">
    <h1>女人</h1>
    <p> •六类食物能有效对抗紫外线 11:15 </p>
    <p> •打造夏美人  受 OL 追捧的清爽发型 10:05 </p>
    <p> •美丽帮帮忙：别让大油脸吓跑男人 09:47 </p>
    <p> •简约雪纺清凉衫  百元搭出欧美范儿 14:51 </p>
    <p> •花边连衣裙超勾人  7 月穿搭出新意 11:04 </p>
  </div>
  <div class="right">
    <h1>健康</h1>
    <p> •女性养生：让女人老得快的 10 个原因 19:18 </p>
    <p> •养生盘点：喝豆浆的九大好处和七大禁忌 09:14</p>
    <p> •养生警惕：14 个护肤心理“错”觉 19:57</p>
    <p> •柿子番茄骨汤  8 种营养师最爱的食物 15:16</p>
    <p> •夏季养生指南：“夫妻菜”宜常吃 10:48 </p>
    <p> •10 条食疗养生方法，居家宅人的养生经 13:54 </p>
  </div>
</div>
</center>
```

```
</body>
</html>
```

在 IE 9.0 中浏览效果如图 19-12 所示，可以看到页面中显示了两个模块，分别是“女人”和“健康”，两者上下排列。

03 添加 CSS 代码，修饰整体样式和 DIV 层。

```
<style>
* {
	padding:0px;
	margin:0px;
}
body {
	font:"宋体";
	font-size:12px;
}
.big{
	width:570px;
	height:210px;
	 border:#C1C4CD 1px solid;
}

</style>
```

在 IE 9.0 中浏览效果如图 19-13 所示，可以看到页面比较原来字体变小，并且大的 DIV 显示了边框。

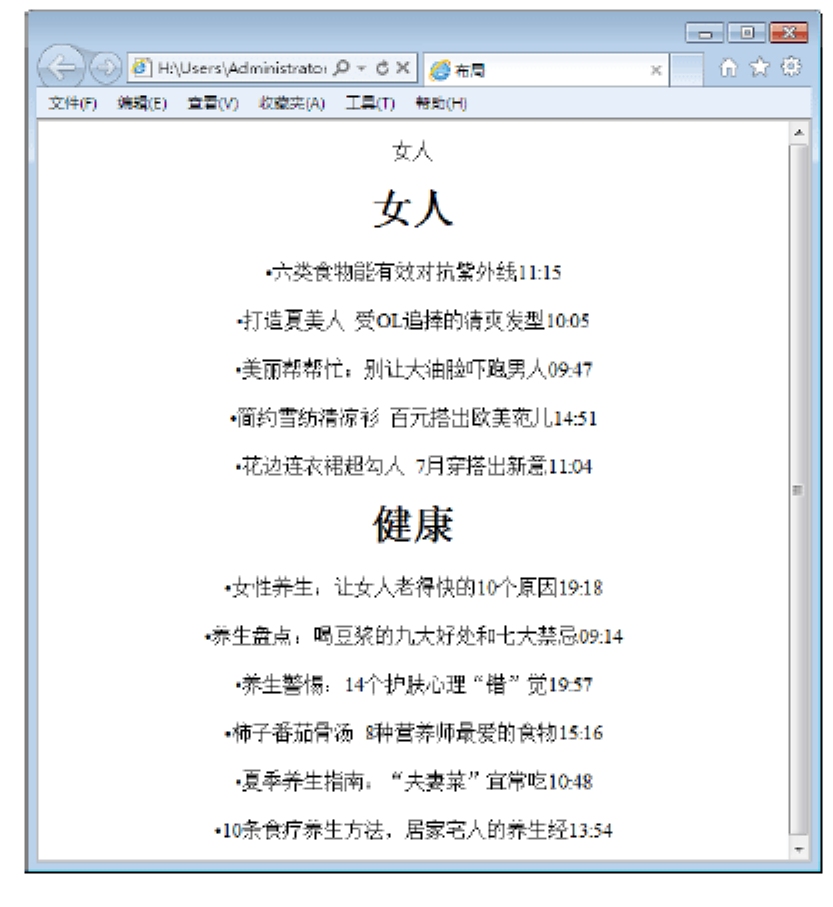


图 19-12　上下排列

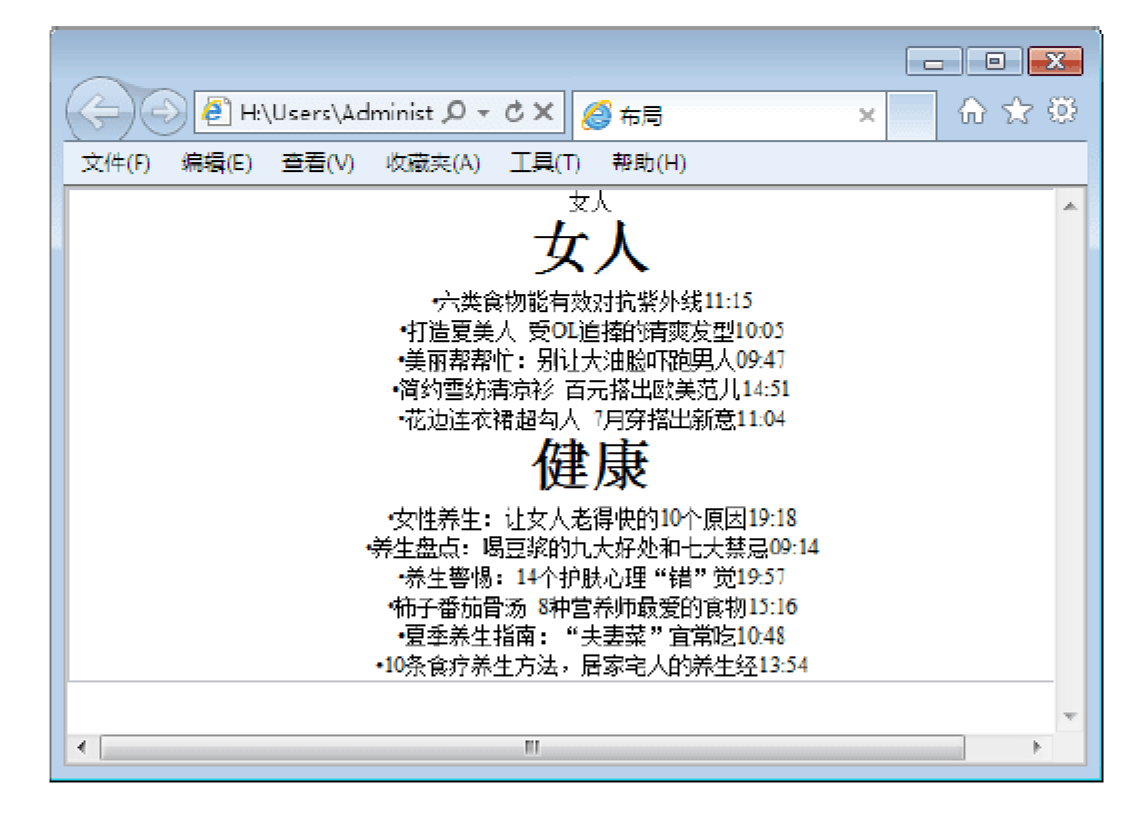


图 19-13　修饰整体样式

04 添加 CSS 代码，设置两个层左右并列显示。

```
.left{
```

```
        width:280px;
        float:right; //设置右边悬浮
        border:#C1C4CD 1px solid;
        }
.right{
        width:280px;
        float:left;//设置左边悬浮
        margin-left:6px;
        border:#C1C4CD 1px solid;
        }
```

在 IE 9.0 中浏览效果如图 19-14 所示，可以看到页面中文本信息左右并列显示，但字体没有发生变化。

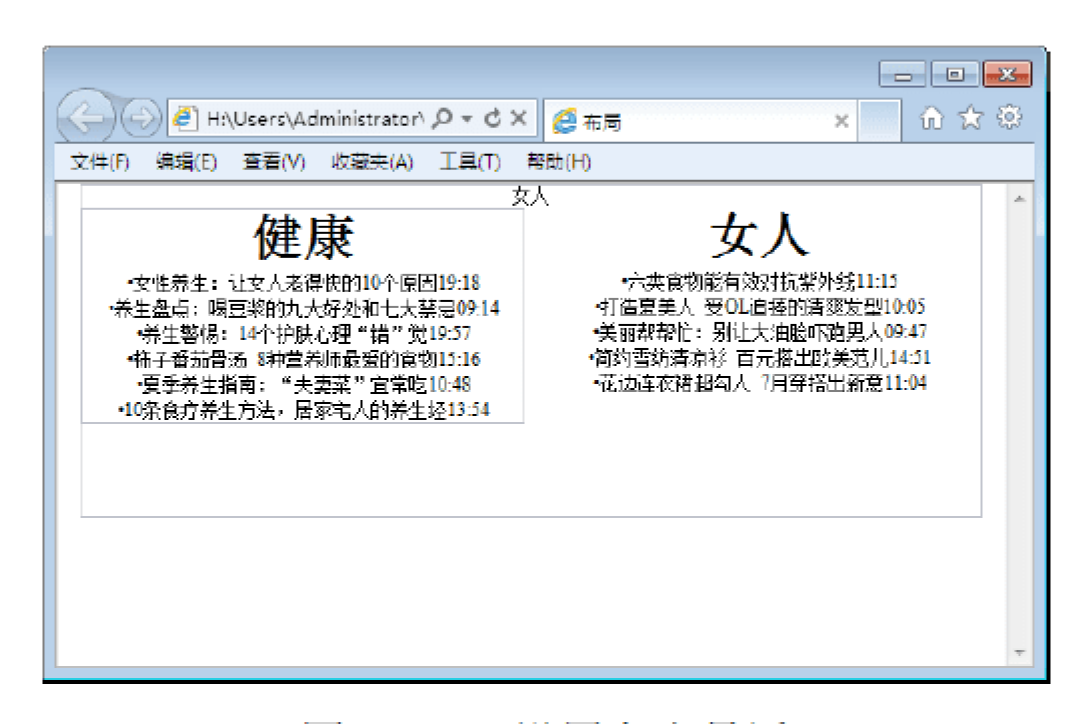


图 19-14　设置左右悬浮

05 添加 CSS 代码，定义文本样式。

```
h1{
        font-size:14px;
        padding-left:10px;
        background-color:#CCCCCC;
        height:20px;
        line-height:20px;
        }
p{
        margin:5px;
        line-height:18px;
         color:#2F17CD;
        }
.pp{
         width:570px;
         text-align:left;
         height:20px;
         background-color:D5E7FD;
```

```
            position:relative;
            left:-3px;
            top:-3px;
            font-size:16px;
            text-decoration:underline;
    }
```

在 IE 9.0 中浏览效果如图 19-15 所示，可以看到页面中文本信息左右并列显示，其字体颜色为蓝色，行高为 18 像素。



图 19-15　文本修饰样式

19.7　专家解惑

1．块元素和行内元素的概念是什么，有哪些特点？

块级元素和行内元素是布局的需要的最基本的两种元素，常见的块级元素有 div、p、form、ul、ol、li 等，常见行内元素有 span、strong、em 等。

块级元素会独占一行，对应于 display:block，可以设置 width、height、margin、padding 属性；行内元素不独占一行，对应于 display:inline，相邻行内元素会排列在同一行，直到排不下才换行，设置 width、height 属性无效，而 margin 和 padding 属性只对设置水平方向的 right 和 left 有效。可以通过修改 display 属性来切换块级元素和行内元素。

display:inline-block 是行内的块级元素，拥有块级元素的特点，可以设置 width、height、margin、padding 值，但又可以和其他行内元素排在同一行。

2．当设置多个 div 并列时，为什么有时会撑破整个布局？

很多时候，尤其是容器内有平行布局，例如两、三个 float 的 div 时，宽度很容易出现问题。在 IE 中，外层的宽度会被内层更宽的 div 挤破。一定要用 Photoshop 或者 Firework 量取像素级的精度。

第 20 章

CSS与JavaScript的搭配应用

网页的吸引人之处，莫过于具有动态效果，利用 CSS 伪类元素可以轻易实现超级链接的动态效果。不过利用 CSS 能实现的动态效果非常有限。在网页设计中，还可以将 CSS 与 JavaScript 结合从而创建出具有动态效果的页面。

20.1 JavaScript 概述

JavaScript 是一种面向对象、结构化、多用途的语言，它支持 Web 应用程序的客户端和服务器方面构件的开发。在客户端利用 JavaScript 脚本语言，可以设计出多种网页特效，从而增加网页浏览量。

20.1.1 JavaScript 概念

JavaScript 是一种脚本编写语言，它采用小程序段的方式实现编程，像其他脚本语言一样，JavaScript 同样也是一种解释性语言，它提供了一个简易的开发过程。JavaScript 是一种基于对象的语言，同时也可以把它看作是一种面向对象语言。这意味着它具有定义和使用对象的能力。因此，许多功能可以由脚本环境中对象的方法与脚本之间进行相互协作来实现。

JavaScript 是动态的，它可以直接对用户或客户输入的数据做出及时响应，无需经过 Web 服务程序。它对用户请求的响应采用以事件驱动的方式进行。在主页中执行某种操作时所产生的动作，就称为“事件”，例如按下鼠标、移动窗口、选择菜单等都可以视为事件。当“事件”被触发时，可能会引起相应的事件响应处理。

JavaScript 具有下面几个特点。

- 一种描述语言，通过浏览器可以直接执行。
- 必须编写在 HTML 文件中，直接查看网页的源码，就可以看到 JavaScript 程序，所以没有保护，任何人都可以通过 HTML 文件复制程序。
- 结构较为自由松散，例如程序中使用变量前，并不需要明确的定义，也不区分大小写。
- 不具有读写文件及网络控制等功能。

20.1.2　JavaScript 的引用

在 HTML 文档中使用 JavaScript 有两种方式，一种是使用<script>标记将将语句嵌入文档，另一种是将 JavaScript 做成独立的文件，引入到 HTML 文档中。

1. 嵌入 JavaScript 代码

如果在 HTML 文件中嵌入 JavaScript 代码，需要在<head>标记中间嵌入一个<script>标记，如下所示。

```
<script language=“JavaScript”>脚本程序</script>
```

此代码语句表示在 HTML 文档里嵌入了一段脚本程序，脚本语言为 JavaScript。中间放置的是脚本程序。

【例 20.1】（实例文件：ch20\20.1.html）

```
<html>
<head>
    <script language = "JavaScript">
        document.write("欢迎来到 JavaScript 动态世界");
   </script>
</head>
<body>
    <P>学习 JavaScript！！！
</body>
</html>
```

该实例功能是在 HTML 文档里输出一个字符串，即“欢迎来到 JavaScript 动态世界”；在 IE 9.0 中浏览效果如图 20-1 所示，可以看到页面中输出了两句话，其中第一句就是 JavaScript 中的输出语句。

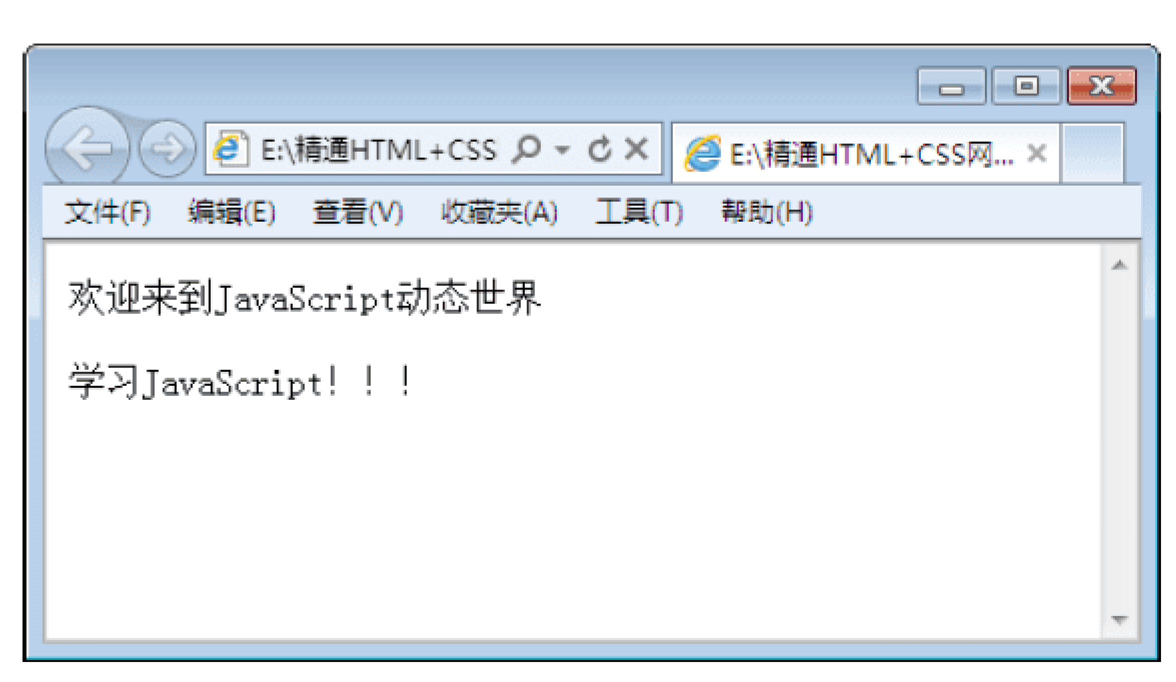


图 20-1　嵌入 JavaScript 代码

在 JavaScript 的语法中，分号（;）是 JavaScript 程序作为一个语句结束的标识符。

2．导入 JavaScript 文件

如果要导入 JavaScript 文件，需要将 JavaScript 做成一个独立文件，然后使用<script>标记将其引入到 HTML 文件中。其引入格式如下所示。

```
<SCRIPT SRC = "hm_19.js"></SCRIPT>
```

【例 20.2】（实例文件：ch20\20.2.html）

```
<html>
<head>
<title>使用外部文件</title>
<script src = "hello.js"></script>
</head>
<body>
<p>此处引用了一个 JavaScript  文件</p>
</body>
</html>
```

在 IE 9.0 中浏览效果如图 20-2 所示，可以看到网页中首先弹出一个对话框，显示提示信息。单击【确定】后，会显示网页内容。

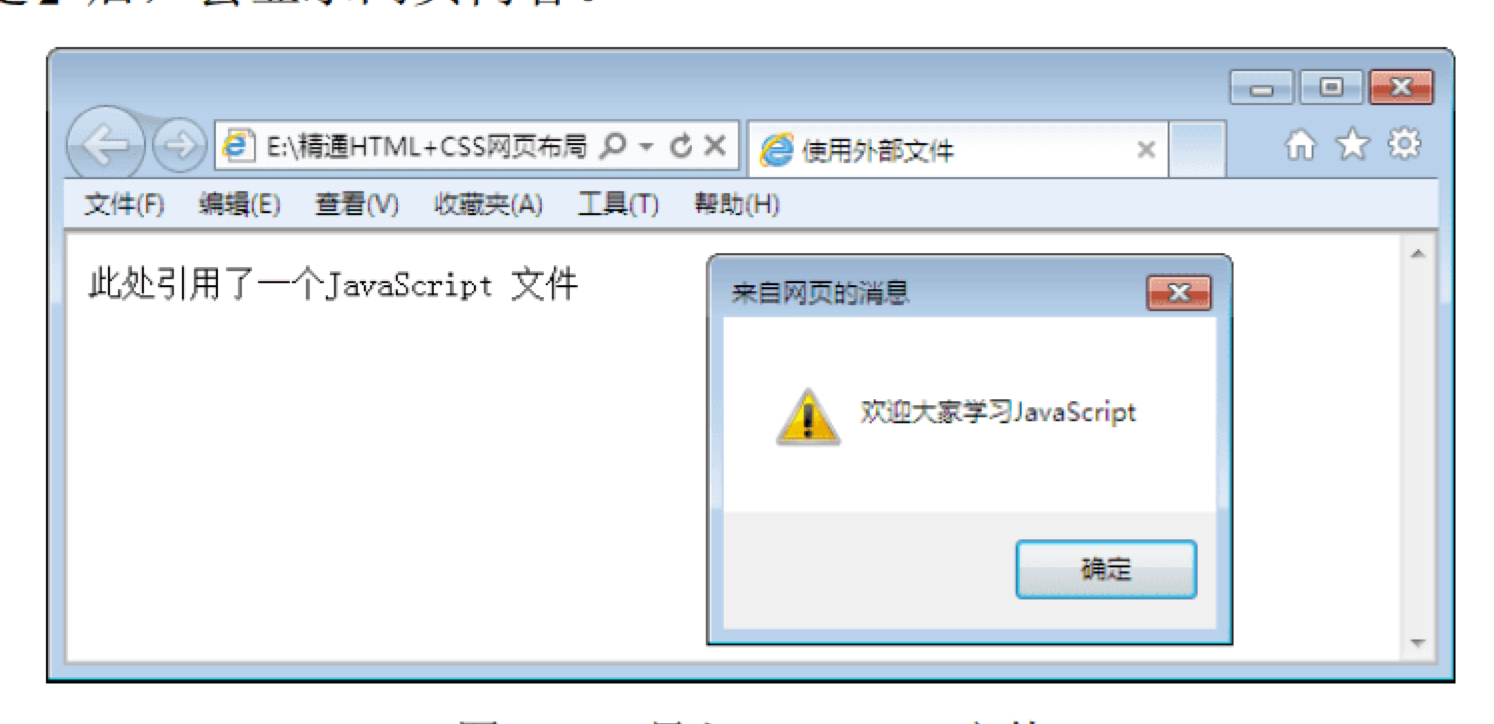


图 20-2　导入 JavaScript 文件

20.1.3　JavaScript 与 CSS

JavaScript 和 CSS 都一个共同的特点，就是两者都是在浏览器上解析并运行的。CSS 用于设置网页上的样式和布局，从而增加网页特效。而 JavaScript 是一种脚本语言，可以直接在网页上被浏览器解释运行。如果将 JavaScript 的程序和 CSS 的静态效果两者结合起来，可以创建出大量的动态特效，即使用 JavaScript 的循环、分支语句来控制 CSS 样式的不断转换。

对于动态特效，实际上前面已经有了实现，例如当鼠标停留在表格中的某一行时，会显示一种颜色，不停留时会显示另外一种颜色。同样，对于其他 HTML 元素，可以用 JavaScript 动态判断，从而调用不同的 CSS 样式。

JavaScript 和 CSS 的结合运用，对于喜爱网页特效的浏览者来说是一大喜讯。作为一个网页设计者，通过对 JavaScript 和 CSS 的学习，可以创作出大量的网页特效。

20.2　JavaScript 语法基础

JavaScript 是一种脚本语言，具有自己的数据类型、变量标识符、运算符和流程控制。但它是一种弱类型语言，便于学习和掌握。

20.2.1　数据类型

在 JavaScript 中有 4 种基本的数据类型：数值（整数和实数）型、字符串型（用“”号或‘’括起来的字符或数值）、布尔型（使 True 或 False 表示）和空值。此外，JavaScript 还定义了其他复合数据类型，例如 Date 对象是一个日期和时间类型。

JavaScript 具有的数据类型如表 20-1 所示。

表 20-1　JavaScript 的基本数据类型

数据类型	数据类型名称	示例
number	数值类型	123、071（十进制）、0X1fa（十六进制）
string	字符串类型	'Hello'、'get the &'、'b@911.com'、"Hello"
object	对象类型	Date、Window、Document、Function
boolean	布尔类型	true 和 false
null	空类型	null
undefined	未定义类型	没有被赋值的变量所具有的“值”

1．数值型

JavaScript 的数值类型可以分为 4 类，即整数、浮点数、内部常量和特殊值。

整数可以为正数、零或者负数；浮点数可以包含小数点，也可以包含一个“e”（大小写均可，在科学记数法中表示“10 的幂”），或者同时包含这两项。整数可以以 10（十进制）、8（八进制）和 16（十六进制）作为基数来表示。

内部常量和特殊值一般不常用，在这里不再介绍。

2．字符串型

字符串是一对单引号（' '）或双引号（" "）和引号中的部分构成的。一个字符串也是 JavaScript 中的一个对象，有专门的属性。

引号中间的部分可以是任意多的字符，如果没有则是一个空字符串。如果要在字符串中使用双引号，则应该将其包含在使用单引号的字符串中，使用单引号时则反之。

3．布尔型

布尔类型 Boolean 表示一个逻辑数值，用于表示两种可能的情况。逻辑真，用 true 表示；逻辑假，用 false 表示。通常，我们使用 1 表示真，0 表示假。

4．undefined

未定义数据类型 undefined 表示变量被创建后，该变量未被赋值，那么此时变量的值就是

未定义数据类型。对于数字，未定义数值表示 NaN；对于字符串，未定义数值表示 undefined；对于逻辑数值，未定义数值表示为假。

5．null

在 JavaScript 里，使用 null 声明的变量并不是 0。null 是一个特殊类型，它表示一个空值，即没有值，而不是 0，0 是有值的。

由于 JavaScript 采用弱类型的形式，因而一个数据的变量或常量不必首先声明，而是在使用或赋值时确定其数据类型。当然也可以先声明该数据类型，它是通过在赋值时自动说明其数据类型的。

20.2.2　变量

在 JavaScript 中使用 var 关键字来声明变量，其语法格式如下所示。

```
var var_name;
```

JavaScript 是一种区分大小写的语言，因此变量 temp 和变量 Temp 代表不同的含义。另外，在命名变量时必须遵循以下规则。

- 变量名由字母、数字、下划线和美元符组成。
- 变量名必须以字母、下划线（_）或美元符（$）开始。
- 变量名不能是保留字。

JavaScript 语言使用等于号（=）给变量赋值，等号左边是变量，等号右边是数值。对变量赋值的语法如下：

```
变量 = 值;
```

JavaScript 里的变量分为全局变量和局部变量两种。其中局部变量就是在函数里定义的变量，在这个函数里定义的变量仅在该函数中有效。如果不加 var，直接对变量进行赋值，那么 JavaScript 将自动把这个变量声明为全局变量。使用示例如下所示。

```
var yourAppleNumber = 100
//等价于
var yourAppleNumber
yourAppleNumber = 100
```

20.2.3　运算符

在 JavaScript 程序中，要完成各种各样的运算是离不开运算符的。运算符用于将一个或几个值进行运算而得出所需要的结果。在 JavaScript 中常用的运算符有算术运算符、比较运算符和逻辑运算符。

1．算术运算符

算术运算符是最简单、最常用的运算符，所以有时也称它们为简单运算符。可以使用它们进行通用的数学计算，如表 20-2 所示。

表 20-2　算术运算符

运算符	说明	示例
+	加法运算符或正值运算符	x+100、100+1000、+100
-	减法运算符或负值运算符	100-60、-100
*	乘法运算符	100*6
/	除法运算符	100/50
%	求模运算符，也就是算术中的求余	100%30
++	将变量值加 1 后再将结果赋值给该变量	x++用于在参与其他运算之前先将自己加 1 后，再用新的值参与其他运算 ++x 用于先用原值与其他运算后，再将自己加 1
--	将变量值减 1 后再将结果赋值给该变量	x--、--x，与++的用法相同

2．比较运算符

比较运算符用于对运算符的两个表达式进行比较，然后根据比较结果返回布尔类型的值，例如，比较两个值是否相同或比较两个数值的大小等。表 20-3 列出了 JavaScript 支持的比较运算符。

表 20-3　比较运算符

运算符	说明	示例
==	判断左右两边表达式是否相等，当左边表达式等于右边表达式时返回 true，否则返回 false	Number == 100 Number1 == Number2
!=	判断左边表达式是否不等于右边表达式，当左边表达式不等于右边表达时返回 true，否则返回 false	Number != 100 Number1 != Number2
>	判断左边表达式是否大于右边表达式，当左边表达式大于右边表达式时返回 true，否则返回 false	Number > 100 Number1 > Number2
>=	判断左边表达式是否大于等于右边表达式，当边表达式大于等于右边表达式时返回 true，否则返回 false	Number >= 100 Number1 >= Number2
<	判断左边表达式是否小于右边表达式，当左边表达式小于右边表达式时返回 true，否则返回 false	Number < 100 Number1 < Number2
<=	判断左边表达式是否小于等于右边表达式，当左边表达式小于等于右边表达式时返回 true，否则返回 false	Number <= 100 Numer <= Number2

3．逻辑运算符

逻辑运算符通常用于执行布尔运算，它们常和比较运算符一起使用来表示复杂比较运算，

这些运算涉及的变量通常不止一个，而且常用于 if、while 和 for 语句中。表 20-4 列出了 JavaScript 支持的逻辑运算符。

表 20-4　逻辑运算符

<table>
<tr><th>运算符</th><th>说明</th><th>示例</th></tr>
<tr><td>&&</td><td>逻辑与，若两边表达式的值都为 true，则返回 true；若任意一个值为 false，则返回 false</td><td>100>60 &&100<200 返回 true
100>50&&10>100 返回 false</td></tr>
<tr><td>||</td><td>逻辑或，只有表达式的值都为 false 时，才返回 false</td><td>100>60||10>100 返回 true
100>600||50>60 返回 false</td></tr>
<tr><td>!</td><td>逻辑非，若表达式的值为 true，则返回 false，否则返 true</td><td>!(100>60)返回 false
!(100>600)返回 true</td></tr>
</table>

除上面介绍的常用运算符之外，JavaScript 还支持条件表达式运算符“?”，这个运算符是三元运算符，它包含 3 个部分：一个计算值的条件和两个根据条件返回的真假值。格式如下所示。

```
条件 ? 表达式 1 : 表达式 2
```

在使用条件运算符时，如果条件为真，则使用表达式 1 的值，否则使用表达式 2 的值。示例如下：

```
( x > y ) ? 100*3 : 11
```

如果 x 的值大于 y 值，则表达式的值为 300；如果 x 的值小于或等于 y 值，表达式的值为 11。

【例 20.3】（实例文件：ch20\20.3.html）

```
<html>
<head>
  <script language = "javascript">
        var a=3;
        var b=5;
        var c=b-a;
         document.write(c+"<br>");
        if(a>b)
               { document.write("a 大于 b<br>");}
        else
               { document.write("a 小于 b<br>");}
       document.write(a>b?"2":"3");
  </script>
</head><body></body>
</html>
```

上面的代码创建了两个变量 a 和 b，变量 c 的值是 b 和 a 的差。下面使用 if 语句判断 a 和

b 的大小，并输出结果。最后使用了一个三元运算符，如果 a>b，则输出 2，否则输出 3。
表示在网页中换行，“+”是一个连接字符串。

在 IE 9.0 中浏览效果如图 20-3 所示，可以看到网页中输出了 JavaScript 语句的执行结果。

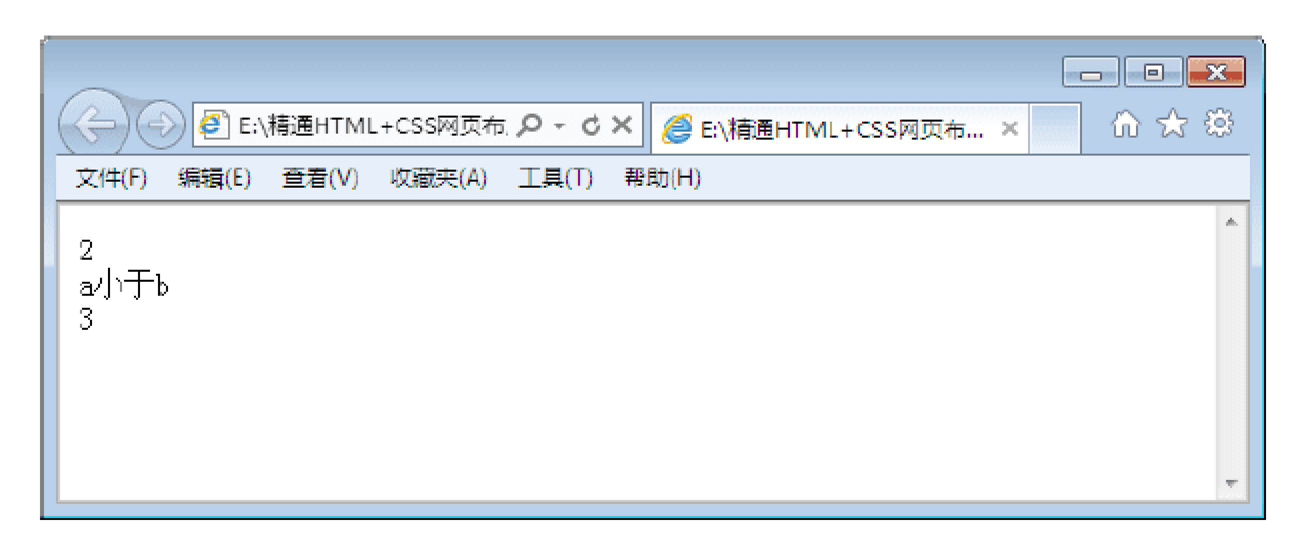


图 20-3 运算符的使用

20.2.4 流程控制语句

JavaScript 语言编程中对流程的控制主要是通过条件判断语句、循环控制语句及 continue、break 来完成的，其中条件判断语句按预先设定的条件执行程序，包括 if 语句和 switch 语句；而循环控制语句则可以重复完成任务，包括 while 语句、do while 语句及 for 语句。

1. if 语句

if 语句是使用最为普遍的条件选择语句，每一种编程语言都有一种或多种形式的 if 语句，在编程中它是经常被用到的。格式如下所示。

```
if(条件语句)
{
    执行语句;
}
```

其中的“条件语句”可以是任何一种逻辑表达式，如果“条件语句”的返回结果为 true，则程序先执行后面大括号中的“执行语句”，然后执行它后面的其他语句。如果“条件语句”的返回结果为 false，则程序跳过“执行语句”，直接执行程序后面的其他语句。大括号的作用就是将多条语句组合成一个复合语句，作为一个整体来处理，如果大括号中只有一条语句，则大括号可以省略。

2. if...else 语句

if...else 语句通常用于一个条件需要两个程序分支来执行的情况。格式如下所示。

```
if(条件语句)
{
    执行语句块 1;
}
else
{
```

```
        执行语句块 2;
    }
```

这种格式在 if 从句的后面添加一个 else 从句，这样当条件语句返回结果为 false 时，执行 else 从句。

【例 20.4】（实例文件：ch20\20.4.html）

```
<html>
<head>
    <script language = "JavaScript">
            var a="john";
            if(a!="john") {
                        document.write("<h1 style='text-align:center;color:red;'>欢迎 JOHN 光临</h1>");
                  }
            else{
                        document.write("<p style='font-size:15px;font-weight:bolder;color:blue'>请重新输入名
                        称</p>");
                  }
   </script>
</head>
<body>
</body>
</html>
```

上面的代码中使用 if...else 语句，对变量 a 的值进行判断，如果 a 值不等于“john”则输出红色标题，否则输出蓝色信息。

在 IE 9.0 中浏览效果如图 20-4 所示，可以看到网页中输出了蓝色信息“请重新输入名称”。

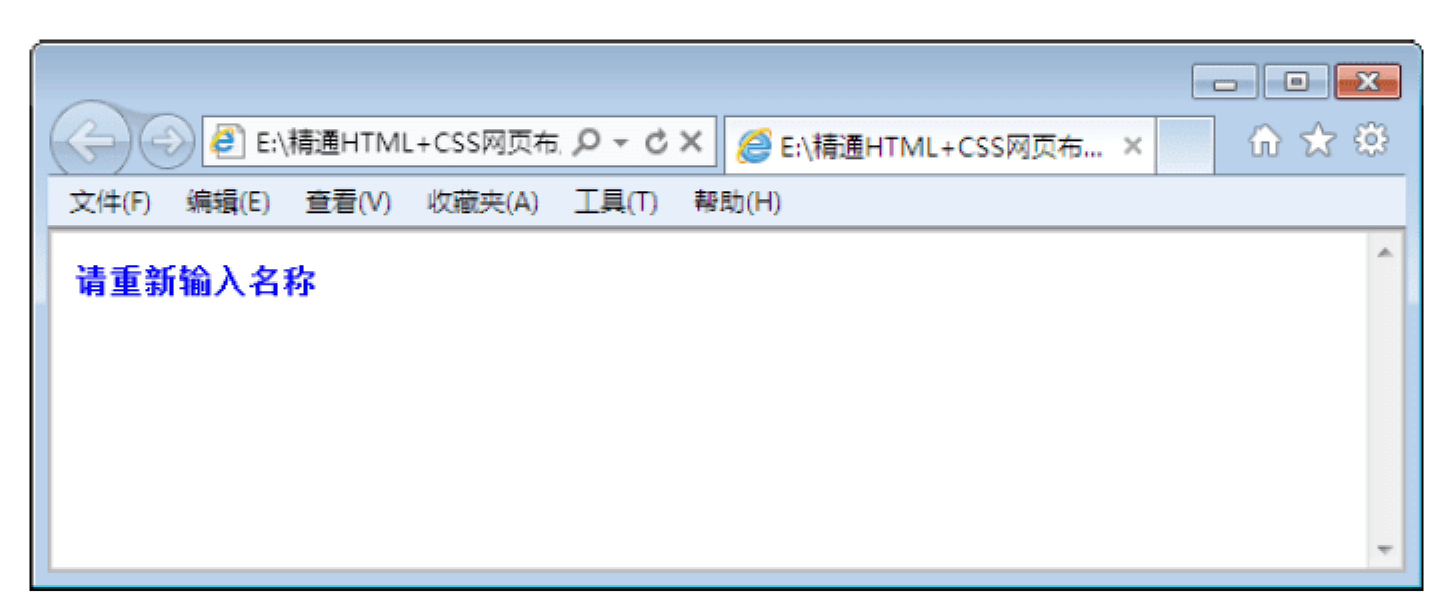


图 20-4　if...else 语句

3．swicth 语句

swicth 选择语句用于将一个表达式的结果同多个值进行比较，根据比较结果选择执行语句。

格式如下所示。

```
switch (表达式)
{
    case 取值 1 :
        语句块 1;break;
    case 取值 2 :
        语句块 2;break;
...
    case 取值 n;
        语句块 n;break;
    default :
        语句块 n+1;
}
```

case 语句只是相当于定义一个标记位置，程序根据 switch 条件表达式的结果，直接跳转到第一个匹配的标记位置处，开始顺序执行后面的所有程序代码，包括后面的其他 case 语句下的代码，直到碰到 break 语句或函数返回语句为止。default 语句是可选的，它匹配上面所有的 case 语句定义的值以外的其他值，也就是前面所有取值都不满足时，就执行 default 后面的语句块。

4. while 语句

while 语句是循环语句，也是条件判断语句。格式如下所示。

```
while(条件表达式语句)
{
  执行语句块
}
```

当“条件表达式语句”的返回值为 true 时，则执行大括号中的语句块，当执行完大括号中的语句块后，再次检测条件表达式的返回值，如果返回值还为 true，则重复执行大括号中的语句块，直到返回值为 false 时，结束整个循环过程，接着往下执行 while 代码段后面的程序代码。

5. do while 语句

do while 语句的功能和 while 语句相似，只不过它是在执行完第一次循环之后才检测条件表达式的值，这意味着包含在大括号中的代码块至少要被执行一次，另外，do while 语句结尾处的 while 条件语句的括号后有一个分号“;”。

格式如下所示。

```
do
{
  执行语句块
}while(条件表达式语句);
```

6．for 语句

for 语句通常由两部分组成，即条件控制部分和循环部分。格式如下所示。

```
for(初始化表达式；循环条件表达式；循环后的操作表达式)
{
  执行语句块
}
```

在使用 for 循环前要先设定一个计数器变量，可以在 for 循环之前预先定义，也可以在使用时直接定义。在上述语法格式中，“初始化表达式”表示计数器变量的初始值；“循环条件表达式”是一个计数器变量的表达式，决定了计数器的最大值；“循环后的操作表达式”表示循环的步长，也就是每循环一次，计数器变量值的变化，该变化可以是增大的，也可以是减小的，或其他运算。for 循环是可以嵌套的，也就是在一个循环里还可以有另一个循环。

【例 20.5】（实例文件：ch20\20.5.html）

```
<html>
<head>
    <script language = "javascript">
                  for(var i=0;i<5;i++){
                                    document.write("<p style='font-size:"+i+"0px'>欢迎学习 javascript</p>");
                     }
   </script>
</head>
<body>
</body>
</html>
```

上面的代码中使用 for 循环输出了不同字体大小的句子。

在 IE 9.0 中浏览效果如图 20-5 所示，可以看到网页中输出了不同大小的句子，从小到大依次排列。

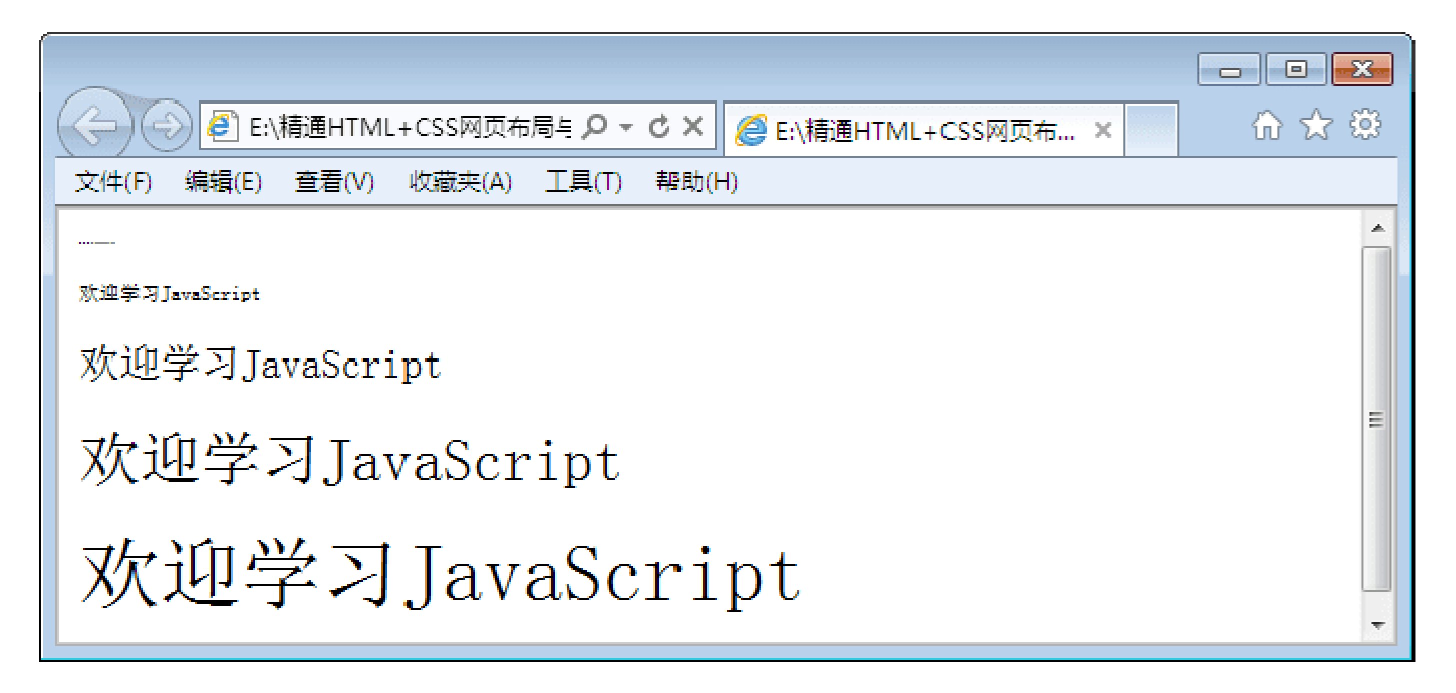


图 20-5　for 循环

除上面的语句之外，JavaScript 还可以使用中断语句 break 和 continue。break 语句可以中

止循环体中的执行语句和 switch 语句。一个无标号的 break 语句会把控制传给当前循环（while、do、for 或 switch）的下一条语句，如果有标号，控制会被传递给当前方法中带有这一标号的循环语句。continue 语句只能出现在循环语句（while、do、for）的循环体语句中，无标号的 continue 语句的作用是跳过当前循环的剩余部分，接着执行下一次循环。

20.2.5 函数

如果在一个程序中需要使用某段功能代码达到十次或更多次，这时可以将这这个功能代码组成一个可以调用的函数，通过调用该函数来执行相应的语句，这样程序就将变得非常简洁，便于后期维护。

在 JavaScript 中定义一个函数时，必须以 function 关键字开头，函数名跟在关键字的后面，接着是函数参数列表和函数所执行的程序代码段。定义一个函数的格式如下所示。

```
function  函数名(参数列表)
{
    程序代码;
    return  表达式;
}
```

在上述格式中，参数列表表示在程序中调用某个函数时传递到函数中的某种类型的值或变量，如果这样的参数多于一个，那么两个参数之间需要用逗号隔开。虽然有些函数并不需要接收任何参数，但在定义函数时也不能省略函数名后面的小括号，保留小括号中的内容为空即可。

另外，函数中的程序代码必须位于大括号之间，如果主程序要求返回一个结果集，就必须使用 return 语句后面跟上这个要返回的结果。当然，return 语句后可以跟一个表达式，返回值将是表达式的运算结果。如果在函数程序代码中省略了 return 语句后的表达式，或者函数结束时没有 return 语句，这个函数就返回一个 undefined 值。

【例 20.6】（实例文件：ch20\20.6.html）

```
<html>
<head>
<title>计算器</title>
<script language="javascript" >
 function compute(op)
  {
     var num1=0;
     var num2=0;
     num1=parsefloat(document.myform.num1.value);
     num2=parsefloat(document.myform.num2.value);
     if (op=="+")
       document.myform.result.value=num1+num2    ;
     if (op=="-")
       document.myform.result.value=num1-num2    ;
```

```
            if (op=="*")
              document.myform.result.value=num1*num2   ;
            if (op=="/" && num2!=0)
              document.myform.result.value=num1/num2   ;
        }
    </script>
    </head>
    <body>
    <form action="" method="post" name="myform" id="myform">
      <p>第一个数
        <input name="num1" type="text" id="num1" size="25">
        <br>
        第二个数
        <input name="num2" type="text" id="num2" size="25">
        </p>
      <p>
        <input name="addbutton" type="button" id="addbutton" value="   +   " onclick="compute('+')">
        <input name="subbutton" type="button" id="subbutton" value="   –   " onclick="compute('-')">
        <input name="mulbutton" type="button" id="mulbutton" value="   ×   " onclick="compute('*')">
        <input name="divbutton" type="button" id="divbutton" value="   ÷   " onclick="compute('/')">
      </p>
      <p>计算结果
        <input name="result" type="text" id="result" size="25">
      </p>
    </form>
    <p>  </p>
    </body>
    </html>
```

在 IE 9.0 中浏览效果如图 20-6 所示，可以看到输入两个不同的数值，可以求它们的和、差、积和商。

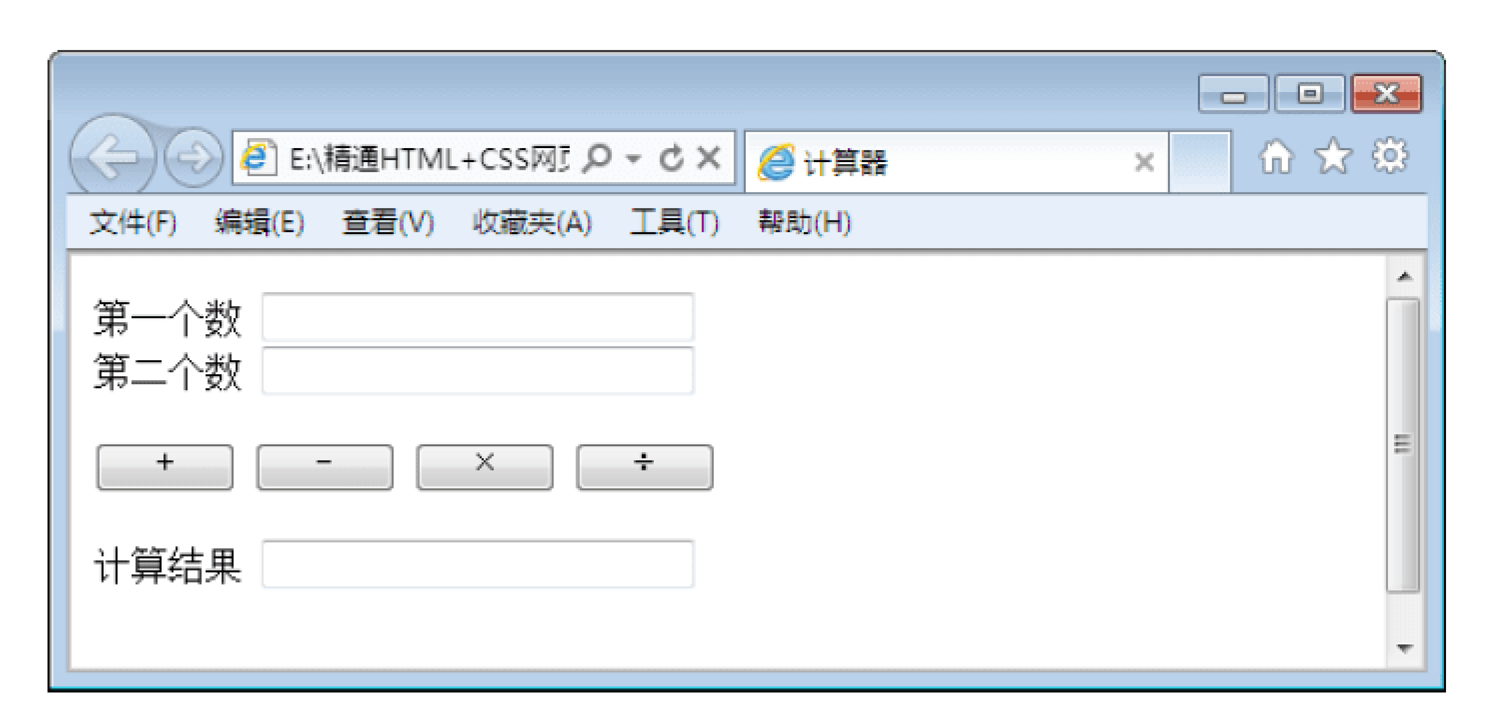


图 20-6　使用函数

JavaScript 不但允许用户根据自己的需要自定义函数，还支持大量的系统函数。常用的系统函数如表 20-5 所示。

表 20-5 常用系统函数

名称	说明
eval()	返回字符串表达式中的值
parseInt()	返回不同进制的数，默认是十进制，用于将一个字符串按指定的进制转换成一个整数
parseFloat()	返回实数，用于将一个字符串转换成对应的小数
escape()	返回对一个字符串进行编码后的结果字符串
encodeURI	返回对一个 URI 字符串编码后的结果
decodeURI	将一个已编码的 URI 字符串解码成最原始的字符串并返回
unescape ()	将一个用 escape 方法编码的结果字符串解码成原始字符串并返回
isNaN()	检测 parseInt()和 parseFloat()函数返回值是否为非数值型，如果是，返回 true，否则，返回 false
abs(x)	返回 x 的绝对值
acos(x)	返回 x 的反余弦值（余弦值等于 x 的角度），用弧度表示
asin(x)	返回 x 的反正弦值
atan(x)	返回 x 的反正切值
atan2(x,y)	返回复平面内点(x，y)对应的复数的幅角，用弧度表示，其值在-π 到 π 之间
ceil(x)	返回大于等于 x 的最小整数
cos(x)	返回 x 的余弦
exp(x)	返回 e 的 x 次幂(ex)
floor(x)	返回小于等于 x 的最大整数
log(x)	返回 x 的自然对数(ln x)
max(a，b)	返回 a，b 中较大的数
min(a，b)	返回 a，b 中较小的数
pow(n，m)	返回 n 的 m 次幂(nm)
random()	返回一个大于 0 小于 1 的随机数
round(x)	返回 x 四舍五入后的值
sin(x)	返回 x 的正弦
sqrt(x)	返回 x 的平方根
tan(x)	返回 x 的正切。isFinite()如果括号内的数字是“有限”的(介于 Number.MIN_VALUE 和 Number.MAX_VALUE 之间）就返回 true；否则返回 false
toString()	把对象转换成字符串。如果在括号中指定一个数值，则转换过程中所有数值转换成特定进制

20.3 综合实例 1——打字效果的文字

文字是网页的灵魂，没有文字的网页，不管特效多么绚丽多彩也没有任何实际意义。文字特效始终是网页设计追求的目标，通过 JavaScript 可以实现多个网页特效。文字的打字效果通过 JavaScript 脚本程序，将预先设置好的文字逐一在页面上显示出来。具体步骤如下所示。

01 分析需求。

如果要在网页中实现打字效果，需要创建预先设置好的文字，作为输出信息。该实例完成后，效果如图 20-7 所示。

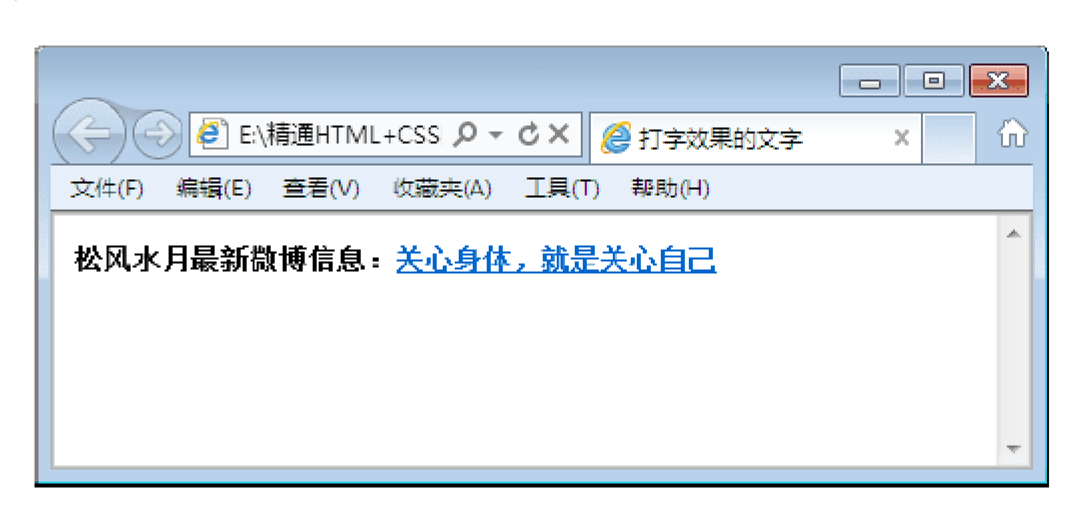


图 20-7 打字效果

02 创建 HTML 页面，设置页面基本样式。

```
<html>
<head>
<title>打字效果的文字</title>
<style type="text/css">
body{font-size:14px;font-weight:bold;}
</style>
</head>
<body>
松风水月最新微博信息：<a id="HotNews" href="" target="_blank"></a>
</body>
</html>
```

上面代码中，在<head>标记中间设置了 body 页面的基本样式，例如字体大小为 14 像素，字形加粗。并在 body 页面中创建了一个超级链接。在 IE 9.0 中浏览效果如图 20-8 所示，可以看到页面中只显示了一个提示信息。

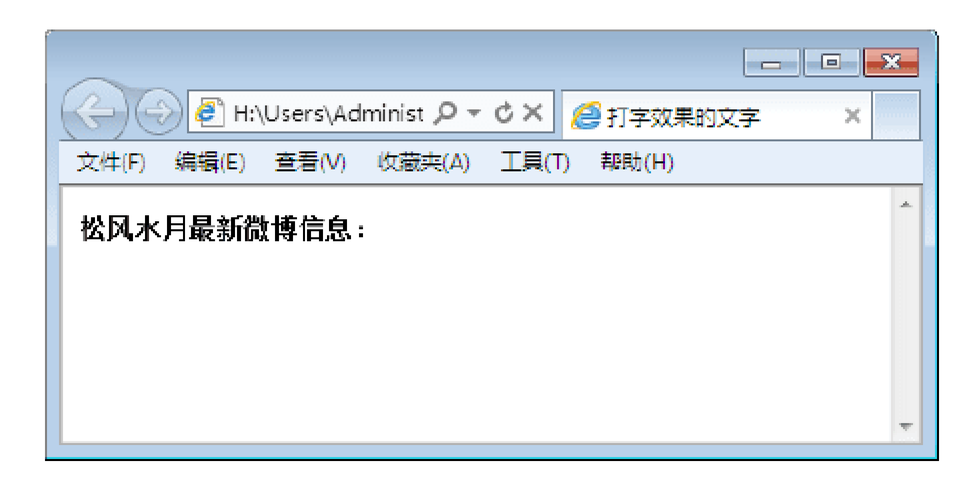


图 20-8 页面基本样式

03 添加 JavaScript 代码，实现打字特效。

```
<script language="JavaScript">
<!--
var NewsTime = 2000;   //每条微博的停留时间
var TextTime = 50;      //微博文字出现的等待时间，越小越快
var newsi = 0;
var txti = 0;
var txttimer;
var newstimer;
var newstitle = new Array();   //微博标题
var newshref = new Array();      //微博链接
newstitle[0] = "健康是身体的本钱";
newshref[0] = "#";
newstitle[1] = "关心身体，就是关心自己";
newshref[1] = "#";
newstitle[2] = "去西藏旅游了";
newshref[2] = "#";
newstitle[3] = "大雨倾盆，很大呀";
newshref[3] = "#";
function shownew()
{
  var endstr = "_"
  hwnewstr = newstitle[newsi];
  newslink = newshref[newsi];
  if(txti==(hwnewstr.length-1)){endstr="";}
  if(txti>=hwnewstr.length){
    clearInterval(txttimer);
    clearInterval(newstimer);
    newsi++;
    if(newsi>=newstitle.length){
      newsi = 0
    }
    newstimer = setInterval("shownew()",NewsTime);
    txti = 0;
    return;
  }
  clearInterval(txttimer);
  document.getElementById("HotNews").href=newslink;
  document.getElementById("HotNews").innerHTML = hwnewstr.substring(0,txti+1)+endstr;
  txti++;
  txttimer = setInterval("shownew()",TextTime);
```

```
}
shownew();
//-->
</script>
```

因为上面的代码是一个整体，这里不分开介绍。代码中主要调用 shownew()函数完成打字效果。在开始部分，定义了多个变量，其中数组对象 newstitle 用于存放文本标题。下面创建了 shownew()函数，并在函数中通过变量和条件获取要显示的文字，通过 setInterval ("shownew()",NewsTime)语句输出文字内容。代码最后使用 shownew()语句循环执行该函数中的输出信息。

在 IE 9.0 中浏览效果如图 20-9 所示，可以看到页面中每隔一段时间，就会在提示信息后逐个打出单个文字，字体颜色为蓝色。

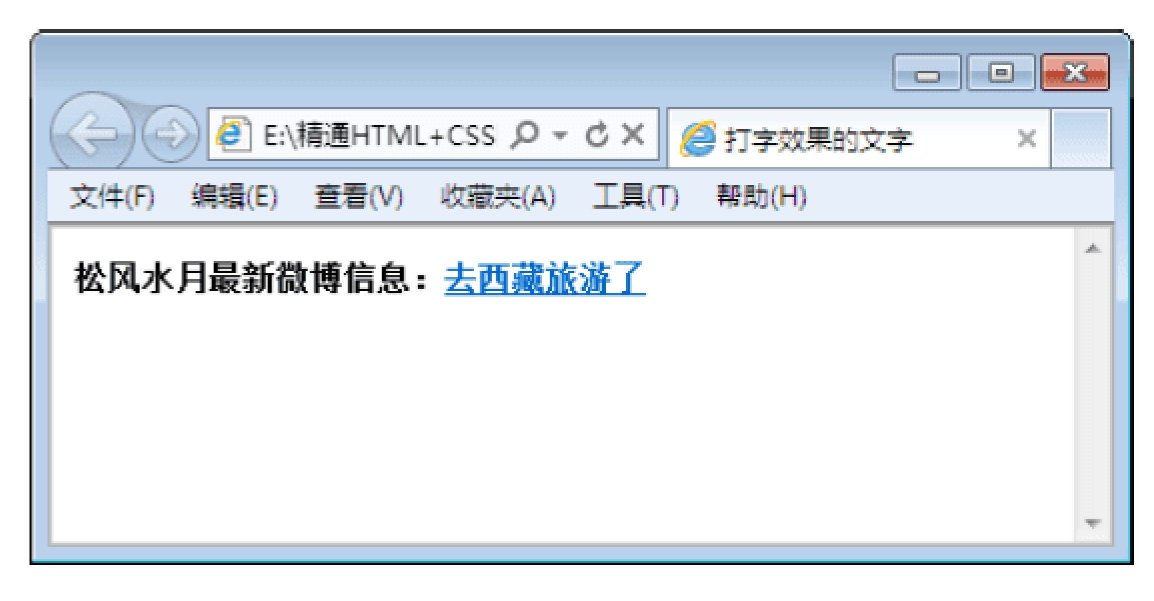


图 20-9　实现打字效果

20.4　综合实例 2——文字升降特效

有的网页为了增加广告宣传力度，往往会设置一些自动升降的文字，用于吸引人的注意。当单击这些升降文字时，会自动跳转到宣传页面。本实例将使用 JavaScript 和 CSS 实现文字升降效果。具体步骤如下所示。

01 分析需求。

如果要实现文字升降，需要指定文字内容和文字升降范围，即在 HTML 页面中指定一个层，用于升降文字。实例完成后，实际效果如图 20-10 所示。

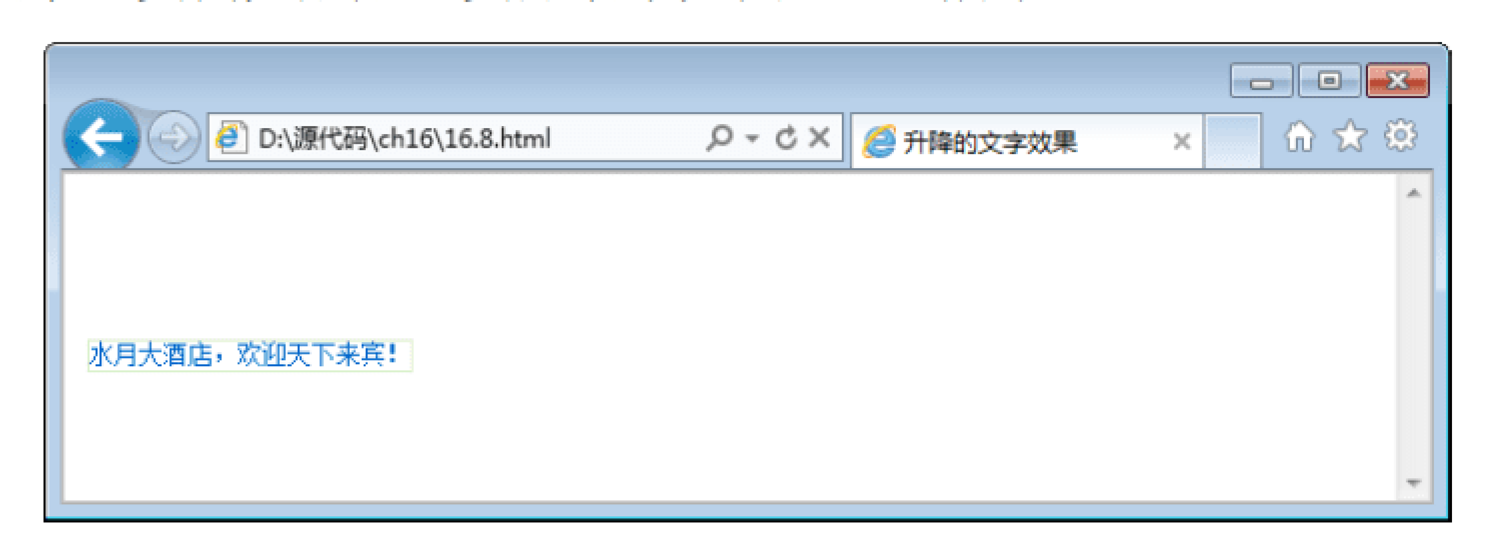


图 20-10　文字升降

02 创建 HTML，构建升降 DIV 层。

```
<html>
<head>
<title>升降的文字效果</title>
</head>
<body>
<div id="napis" style="position: absolute;top: -50;color: #000000;font-family:宋体;font-size:9pt;border:1px
#ddeecc solid">
<a href="" style="font-size:12px;text-decoration:none;">
水月大酒店，欢迎天下来宾！
</a></div>
<script language="JavaScript">
<!--
setTimeout('start()',20);
//-->
</script>
</body>
</html>
```

上面的代码创建了一个 DIV 层，用于存放升降的文字，层 id 的名称是 napis，并在层的 style 属性中定义了层显示样式，如字体大小、带有边框、字形等。在 DIV 层中，创建了一个超级链接，并设定了超级链接的样式。其中的 script 代码，用于定时调用 start 函数。

在 IE 9.0 中浏览效果如图 20-11 所示，可以看到页面空白，无文字显示。

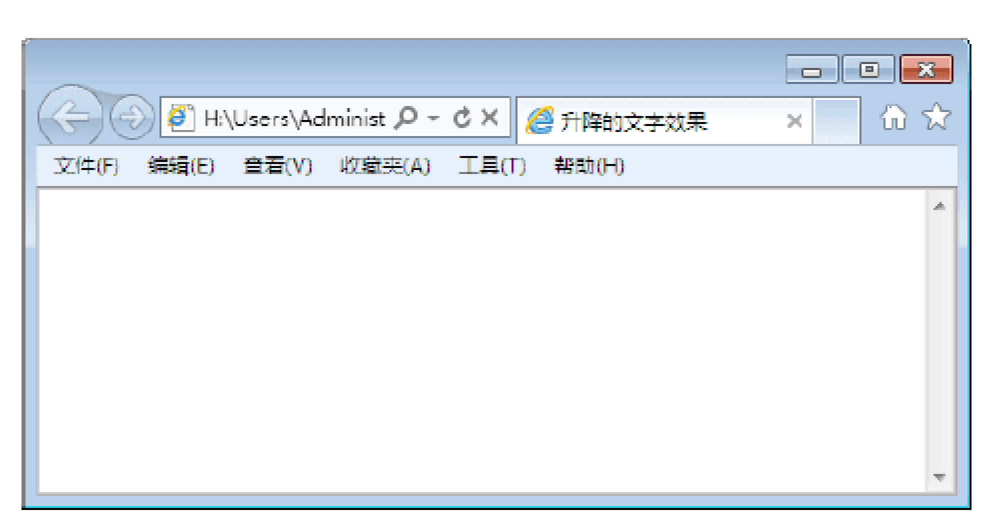


图 20-11　空白页面效果

03 添加 JavaScript 代码，实现文字升降。

```
<script language="JavaScript">
<!--
done = 0;
step = 4
function anim(yp,yk)
{
    if(document.layers) document.layers["napis"].top=yp;
    else document.all["napis"].style.top=yp;
    if(yp>yk) step = -4
    if(yp<60) step = 4
```

```
        setTimeout('anim('+(yp+step)+','+yk+')', 35);
    }
    function start()
    {
        if(done) return
        done = 1;
        if(navigator.appName=="Netscape") {
        var nap=document.getElementById("napis");
        nap.left=innerWidth/2 - 145;
        anim(60,innerHeight - 60)
    }
    else {
        napis.style.left=11;
        anim(60,document.body.offsetHeight - 60)
    }//-->
    </script>
```

上面的代码创建了函数 anim()和 start()，其中 anim()函数用于设定每次升降的数值，start()函数用于设定每次开始的升降坐标。在 IE 9.0 中浏览效果如图 20-12 所示，可以看到页面中超级链接自动上下移动。

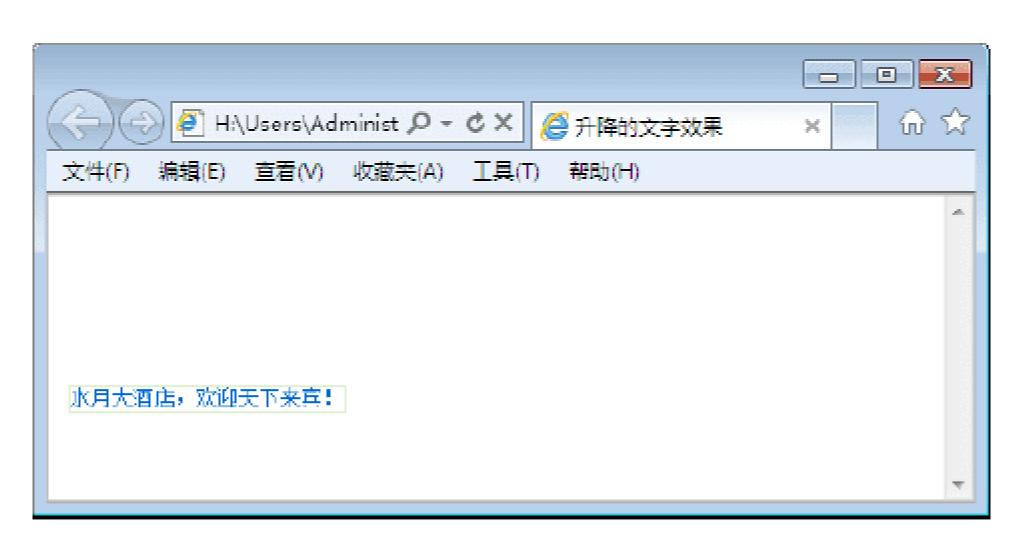


图 20-12　上下移动效果

20.5　综合实例 3——跑马灯效果

网页中有一种特效称为跑马灯，即文字从左到右自动输出，和写字楼的广告霓虹灯非常相似。在网页中，如果 CSS 样式非常完美，就能设计出更加靓丽的网页效果。具体步骤如下所示。

01 分析需求。

要完成跑马灯效果，需要使用 JavaScript 语言设置文字内容、移动速度和相应的输入框，使用 CSS 设置显示文字样式。输入框用来显示水平移动文字。完成后，实际效果如图 20-13 所示。

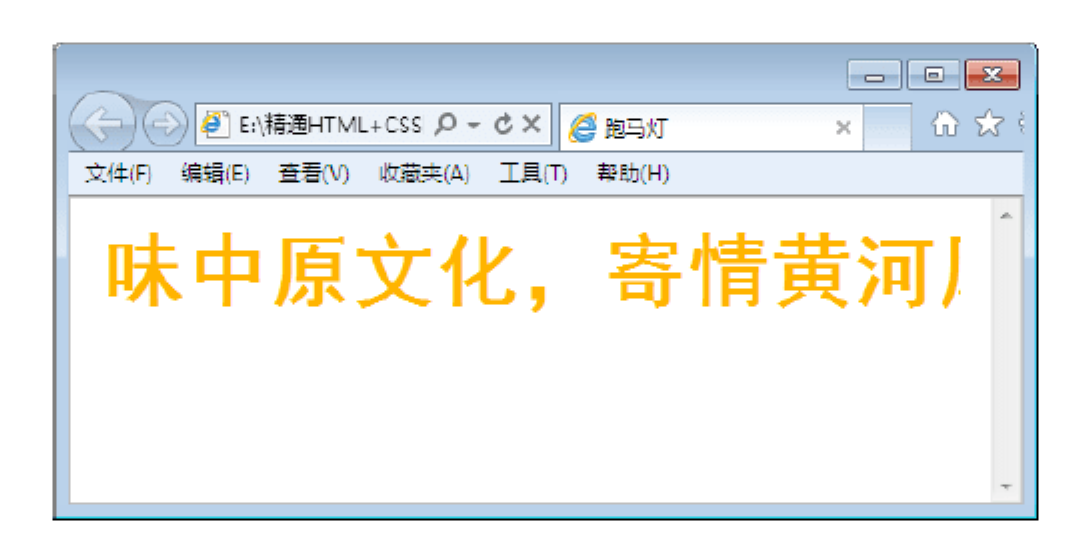


图 20-13 跑马灯效果

02 创建 HTML，实现输入表单。

```
<html>
<head>
<title>跑马灯</title>
</head>
<body onLoad="LenScroll()">
<center>
<form name="nextForm">
<input type=text name="lenText">
</form>
</center>
</body>
```

上面的代码非常简单，创建一个表单，表单中存放了一个文本域，用于显示移动文字。

在 IE 9.0 中浏览效果如图 20-14 所示，可以看到页面中只存在一个文本域，没有其他显示信息。

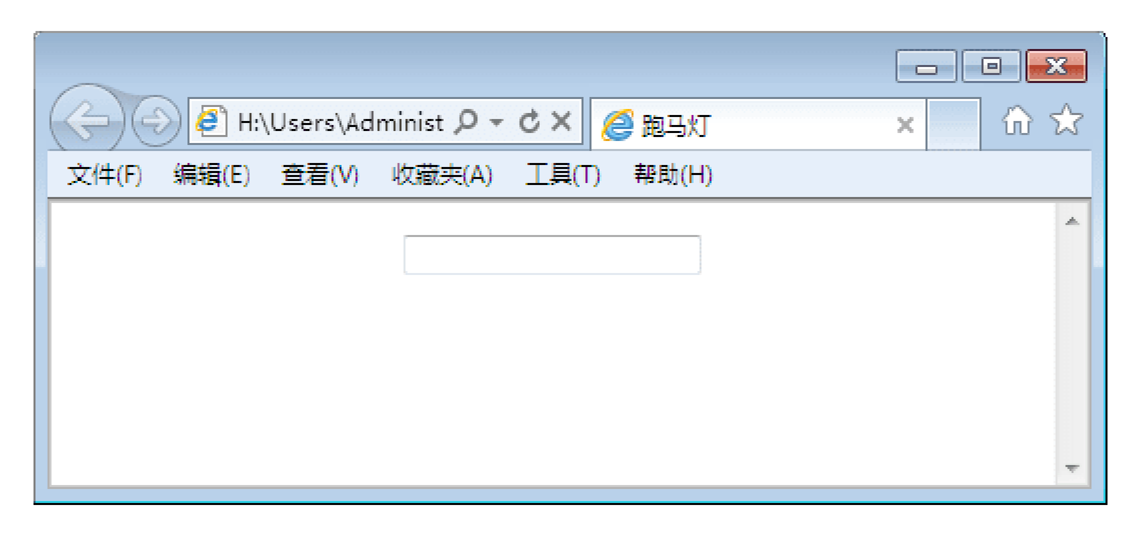


图 20-14 实现基本表单

03 添加 JavaScript 代码，实现文字移动。

```
<script language="javascript">
var msg="品味中原文化，寄情黄河风景";          //移动文字
var interval = 400;                            //移动速度
var seq=0;
function LenScroll() {
  document.nextForm.lenText.value = msg.substring(seq, msg.length) + "      " + msg;
  seq++;
```

```
    if ( seq > msg.length )
      seq = 0;
    window.setTimeout("LenScroll();", interval);
  }
  </script>
```

在上面的代码中，变量 msg 用于定义移动的文字内容，变量 interval 用于定义文字移动速度，LenScroll()函数用于在表单输入框中显示移动信息。

在 IE 9.0 中浏览效果如图 20-15 所示，可以看到输入框中显示了移动信息，并且文字从右向左移动。

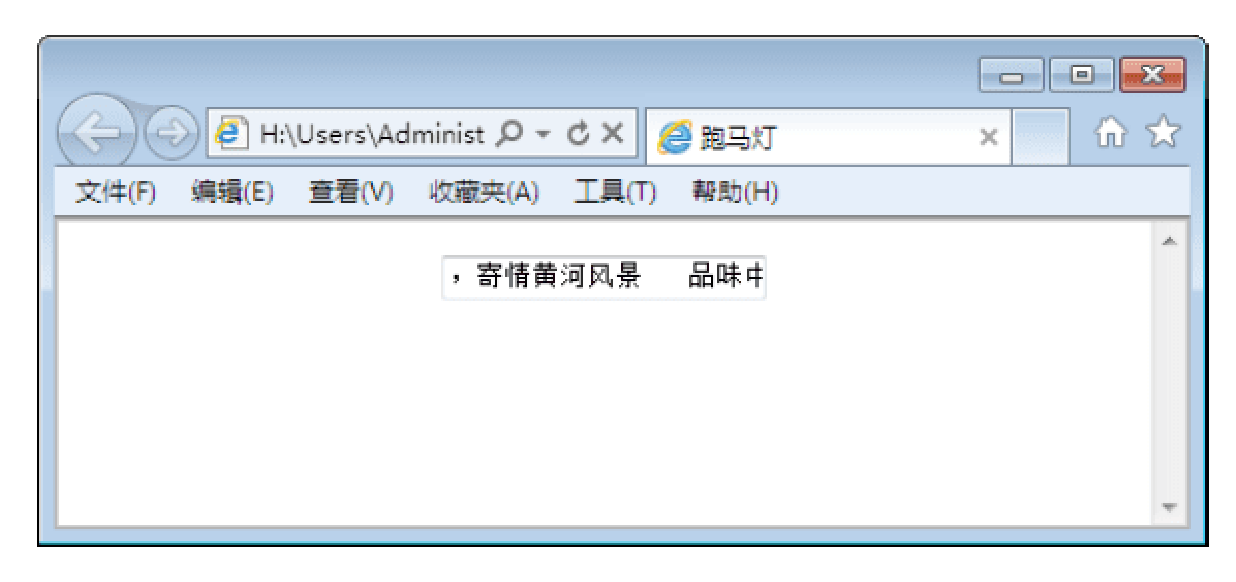


图 20-15 实现移动效果

04 添加 CSS 代码，修饰输入框和页面。

```
<style type="text/css">
<!--
body{
  background-color:#FFFFFF;           /* 页面背景色 */
}
input{
  background:transparent;             /* 输入框背景透明 */
  border:none;                        /* 无边框 */
  color:#ffb400;
  font-size:45px;
  font-weight:bold;
  font-family:黑体;
}--></style>
```

上面的代码设置了页面背景颜色为白色，在 input 标记选择器中，定义了边框背景为透明，无边框，字体颜色为黄色，字体大小为 45 像素，加粗并黑体显示。在 IE 9.0 中浏览效果如图 20-16 所示，可以看到与原来的页面相比，显示字体变大，颜色为黄色，没有输入框显示。

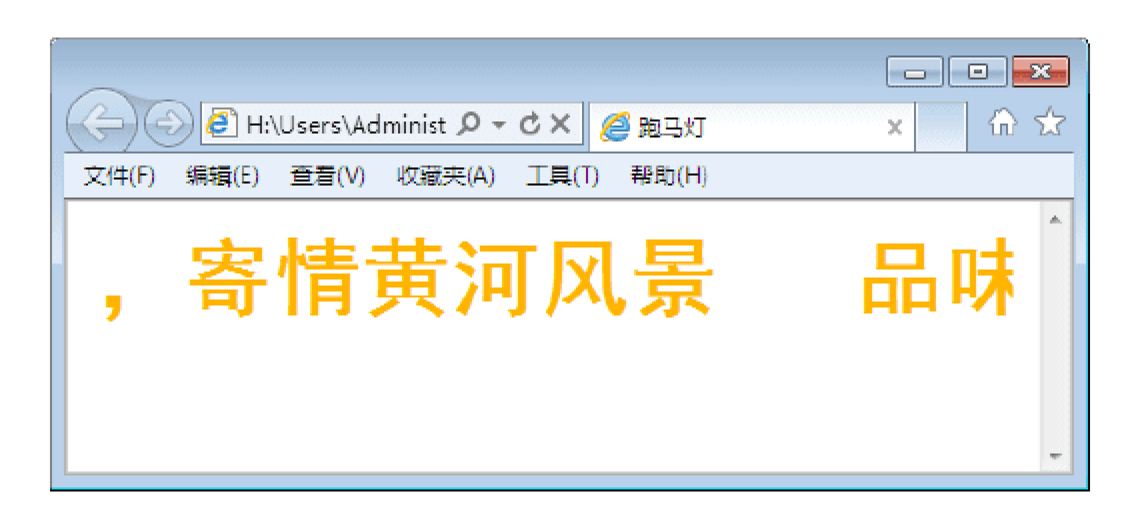


图 20-16　最终效果

20.6　综合实例 4——闪烁图片

图片闪烁是一种常用的特效，用 JavaScript 实现起来非常简单，需要注意时间间隔这个参数，数值越大闪烁越不连续，数值越小闪烁越厉害，可以随意更改这个值，直到取得满意的效果为止。具体步骤如下所示。

01 分析需求。

将图片放在一个 DIV 层上，设定图片为可见，然后使用 JavaScript 程序代码设置 DIV 层的显示和隐藏，这样就获得了图片的闪烁效果。实例完成后，效果如图 20-17 所示。

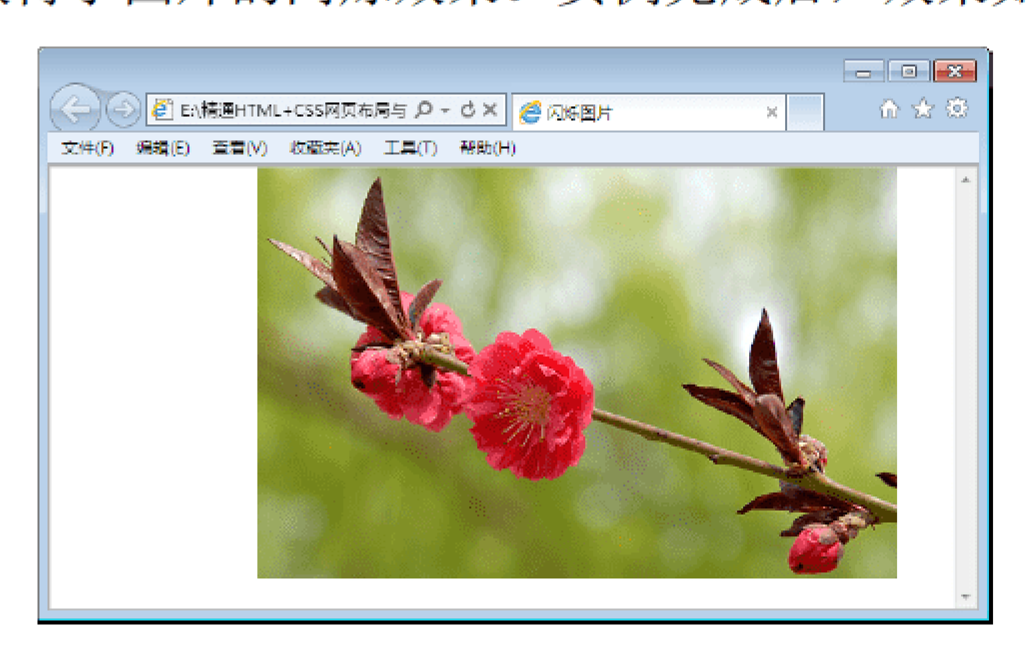


图 20-17　图片闪烁效果

02 创建 HTML 页面，构建 DIV 层。

```
<html>
<head>
<title>闪烁图片</title>
</head>
<body onload="socceronload()" topmargin="0">
<div id="soccer" style="position:absolute; left:150; top:0">
<a href="">
<img src="feng.jpg" border="0"></a>
</div>
</body>
</html>
```

在上面的代码中，创建了一个层，其 ID 名称为 soccer，样式为绝对定位，坐标位置为（150,0）。然后在层中创建了一个图片，不带有边框。

在 IE 9.0 中浏览效果如图 20-18 所示，可以看到页面中显示一个图片，不具有闪烁效果。

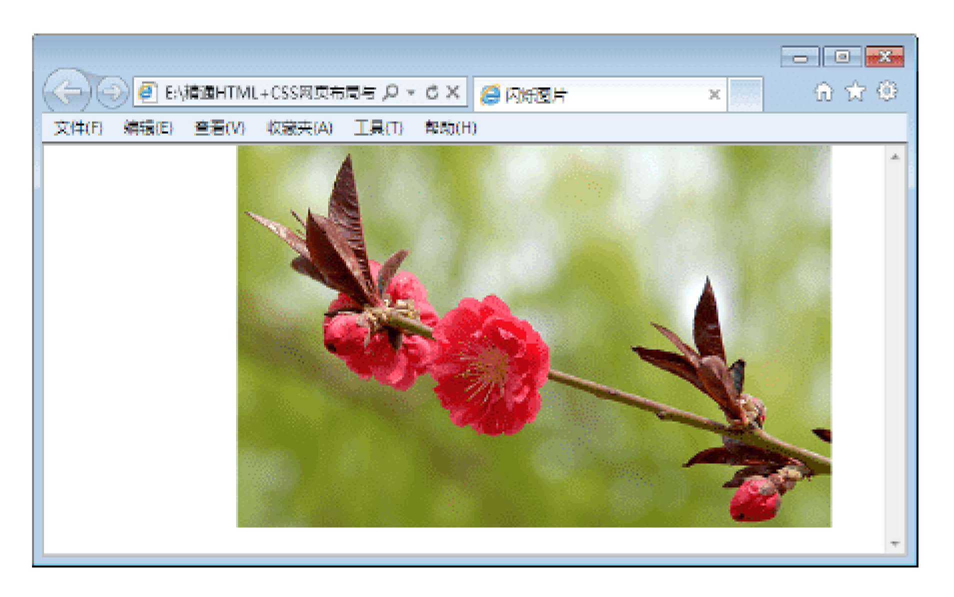


图 20-18　创建图片

03 添加 JavaScript 代码，实现图片闪烁效果。

```
<script language="JavaScript">
var msecs = 500; //改变时间得到不同的闪烁间隔；
var counter = 0;
function soccerOnload() {
setTimeout("blink()", msecs);
}
function blink() {
(soccer.style.visibility == "hidden") ? "visible" : "hidden";
counter +=1;
setTimeout("blink()", msecs);
}
</script>
```

在 JavaScript 代码中，创建变量 msecs 用于定义闪烁时间间隔，创建变量 counter 用于计数。在函数 soccerOnload()中设定每隔指定时间图片闪烁一次，函数 blink()用于设定图片显示，即图片是隐藏还是可见。

在 IE 9.0 中浏览效果如图 20-19 所示，可以看到网页中显示一个图片，在指定时间内闪烁，图中是闪烁消失的时刻。

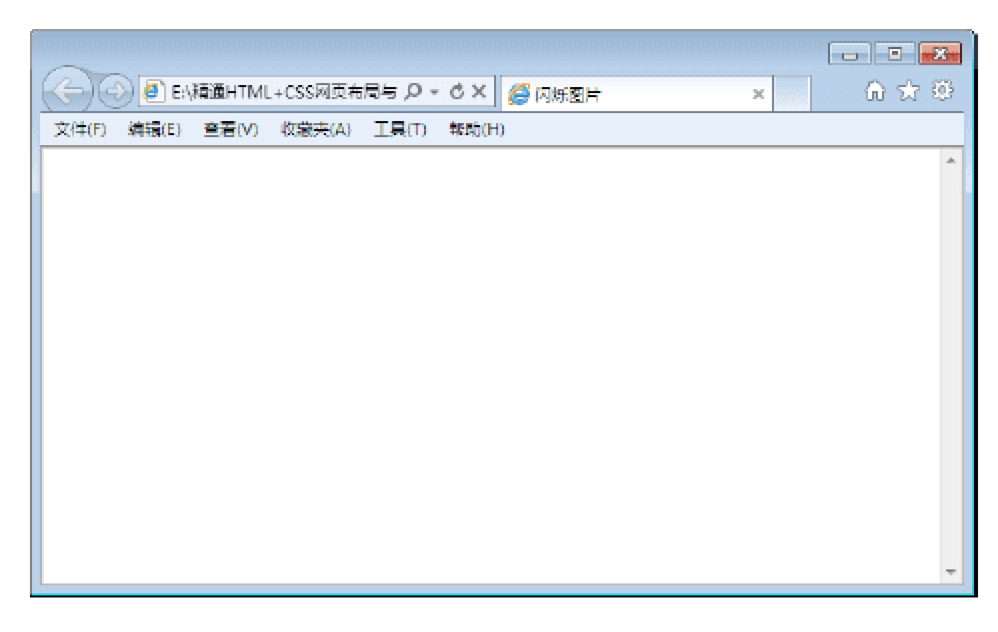


图 20-19　最终效果

20.7 综合实例 5——左右移动的图片

在网页中，经常会出现从右到左移动或者从左到右移动的一张或者多张图片。这样不但可以增加页面效果，也能获取经济利益。本实例将使用 JavaScript 和 CSS 创建一个左右移动的图片。具体步骤如下所示。

1. 分析需求

要实现左右移动的图片，需要在页面上定义一张图片，然后利用 JavaScript 程序代码，获取图片对象，并使其在一定范围内，即水平方向上自由移动。实例完成后，效果如图 20-20 所示。

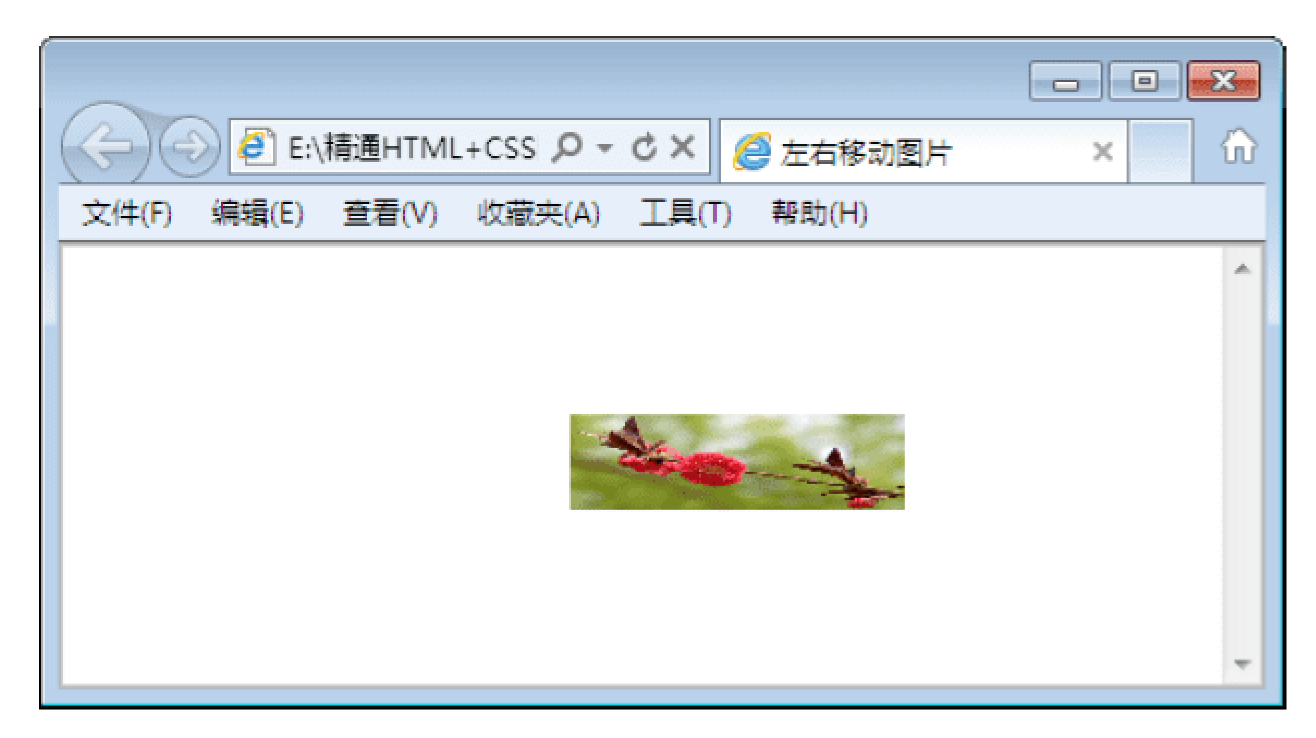


图 20-20 图片移动效果

2. 创建 HTML 页面，导入图片

```
<html>
<head>
<title>左右移动图片</title>
</head>
<body>
<img src="feng.jpg" name="picture"
style="position: absolute; top: 70px; left: 30px;" BORDER="0" WIDTH="140" HEIGHT="40">
<script LANGUAGE="JavaScript"><!--
setTimeout("moveLR('picture',300,1)",10);
//--></script>
</body>
</html>
```

上面的代码中，定义了一个图片，图片是绝对定位，左边坐标位置是（70,30），无边框，宽度为 140 像素，高度为 40 像素。在 script 标记中使用 setTimeout 方法，定时移动图片。

在 IE 9.0 中浏览效果如图 20-21 所示，可以看到网页中显示一个图片。

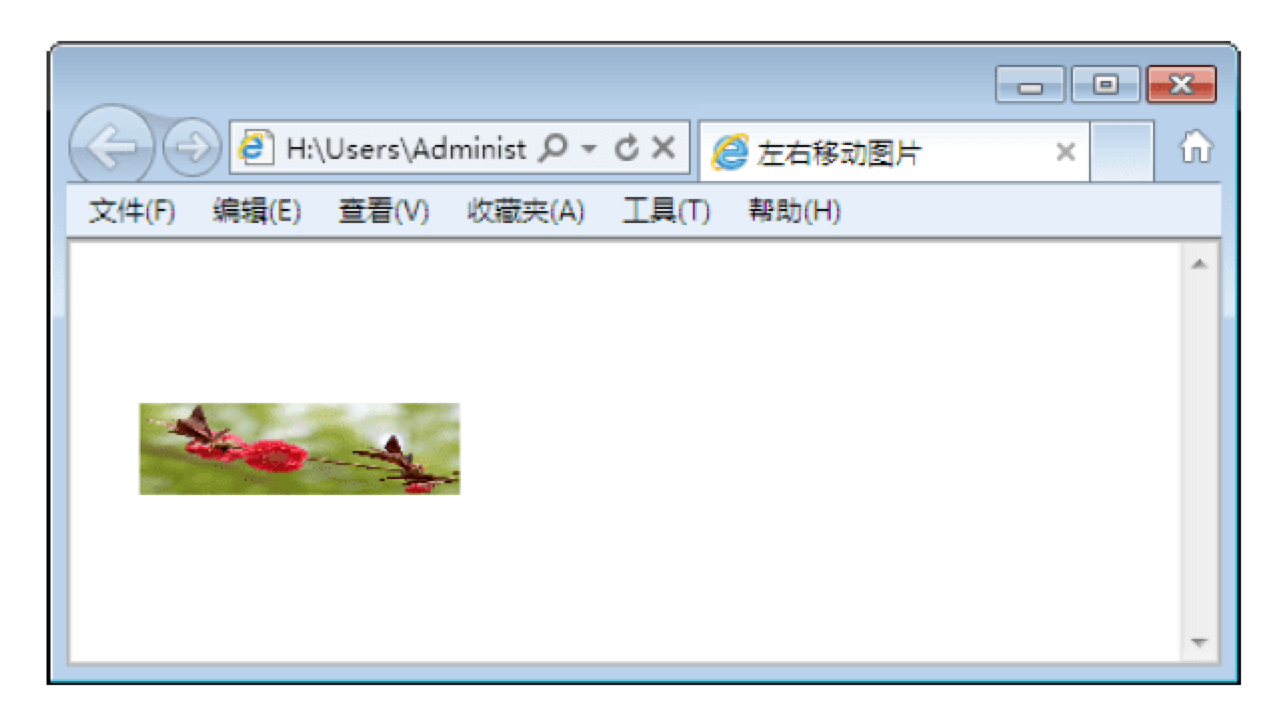


图 20-21 图片显示

3．加入 JavaScript 代码，实现图片的左右移动

```
<script LANGUAGE="JavaScript"><!--
step = 0;
obj = new Image();
function anim(xp,xk,smer) //smer = direction
{
obj.style.left = x;
x += step*smer;
if (x>=(xk+xp)/2) {
if (smer == 1) step--;
else step++;
}
else {
if (smer == 1) step++;
else step--;
}
if (x >= xk) {
x = xk;
smer = -1;
}
if (x <= xp) {
x = xp;
smer = 1;
}
// if (smer > 2) smer = 3;
setTimeout('anim('+xp+','+xk+','+smer+')', 50);
}
function moveLR(objID,movingarea_width,c)
{
if (navigator.appName=="Netscape") window_width = window.innerWidth;
else window_width = document.body.offsetWidth;
```

```
obj = document.images[objID];
image_width = obj.width;
x1 = obj.style.left;
x = Number(x1.substring(0,x1.length-2)); // 30px -> 30
if (c == 0) {
if (movingarea_width == 0) {
right_margin = window_width - image_width;
anim(x,right_margin,1);
}
else {
right_margin = x + movingarea_width - image_width;
if (movingarea_width < x + image_width) window.alert("No space for moving!");
else anim(x,right_margin,1);
}
}
else {
if (movingarea_width == 0) right_margin = window_width - image_width;
else {
x = Math.round((window_width-movingarea_width)/2);
right_margin = Math.round((window_width+movingarea_width)/2)-image_width;
}
anim(x,right_margin,1);
}
}
//--></script>
```

上面的代码和文字水平方向移动原理基本相同，只是对象不同，这里就不再介绍。

在 IE 9.0 中浏览效果如图 20-22 所示，可以看到网页中显示一个图片，在水平方向上自由移动。

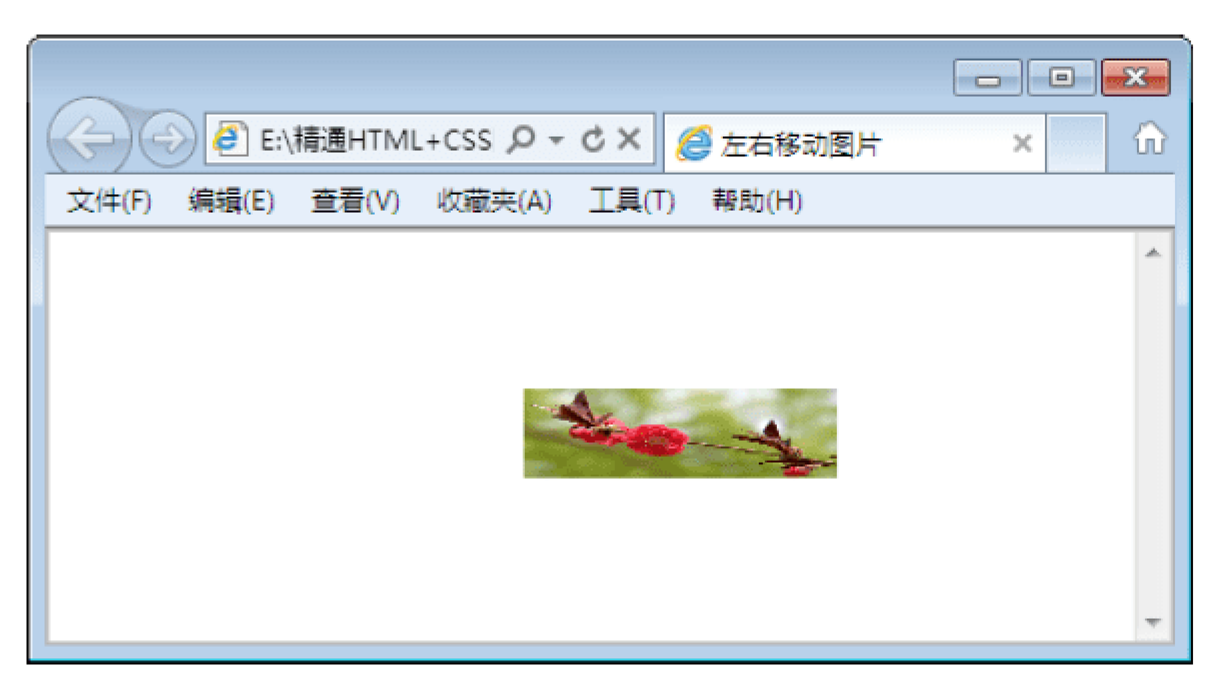


图 20-22　最终效果

20.8 综合实例 6——向上滚动菜单

网页包含信息比较多的时候，就需要设计一些导航菜单来实现页面导航。使用 JavaScript 代码将菜单做成动态效果，菜单会更加吸引人。本实例将结合前面学习的内容，创建一个向上滚动的菜单。具体步骤如下所示。

01 分析需求。

要实现菜单自动从下到上滚动，需要把握两个元素，一个是使用 JavaScript 实现要滚动的菜单，即导航栏，另一个是使用 JavaScript 控制菜单移动方向。实例完成后，效果如图 20-23 所示。

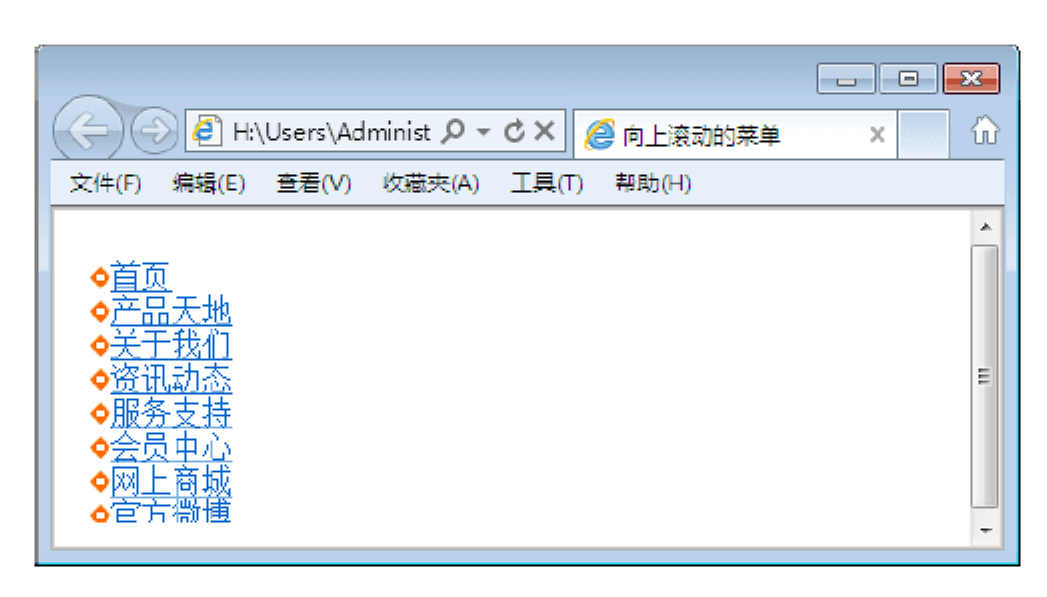


图 20-23 滚动菜单

02 构建 HTML 页面。

```
<html>
<head>
<title>向上滚动的菜单</title>
</head>
<body bgcolor="#FFFFFF" text="#000000">
</body>
</html>
```

上面的代码比较简单，只实现了一个空白页面，页面背景色为白色，前景色为黑色。在 IE 9.0 中浏览效果如图 20-24 所示，可以看到一个空白页面。

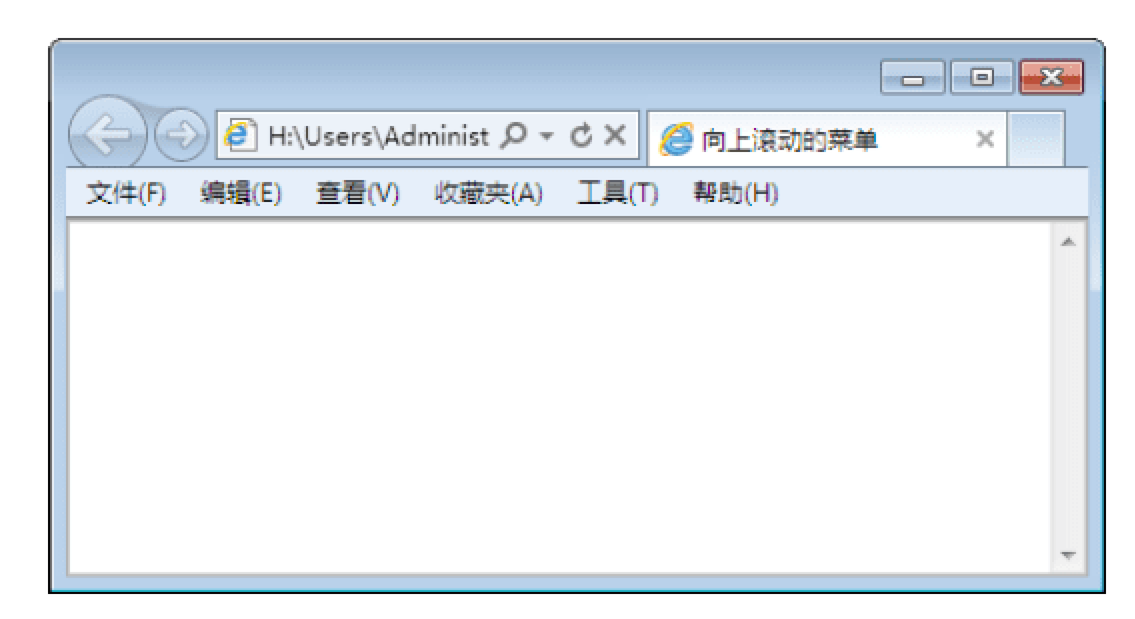


图 20-24 空白 HTML 页面

03 添加 JavaScript 代码，实现菜单滚动。

```
<script language=javascript>
<!--
  var index = 9
  link = new Array(8);
  link[0] ='time1.htm'
  link[1] ='time2.htm'
  link[2] ='time3.htm'
  link[3] ='time1.htm'
  link[4] ='time2.htm'
  link[5] ='time3.htm'
  link[6] ='time1.htm'
  link[7] ='time2.htm'
  link[8] ='time3.htm'
  text = new Array(8);
  text[0] ='首页'
  text[1] ='产品天地'
  text[2] ='关于我们'
  text[3] ='资讯动态'
  text[4] ='服务支持'
  text[5] ='会员中心'
  text[6] ='网上商城'
  text[7] ='官方微博'
  text[8] ='企业文化'
  document.write ("<marquee scrollamount='1' scrolldelay='100' direction= 'up' width='150' height='150'>");
  for (i=0;i<index;i++)
  {
    document.write(" <img src='dian3.gif' width='12' height='12'><a href="+link[i]+" target='_blank'>");
    document.write (text[i] + "</A><br>");
  }
  document.write ("</marquee>")
// --></script>
```

上面的代码创建了两个数组对象 link 和 text，用来存放菜单链接对象和菜单内容，在下面的 JavaScript 代码中，使用<marquee>定义页面在垂直方向上下移动。

在 IE 9.0 中浏览效果如图 20-25 所示，可以看到面左侧有一个菜单，自下向上自由移动。

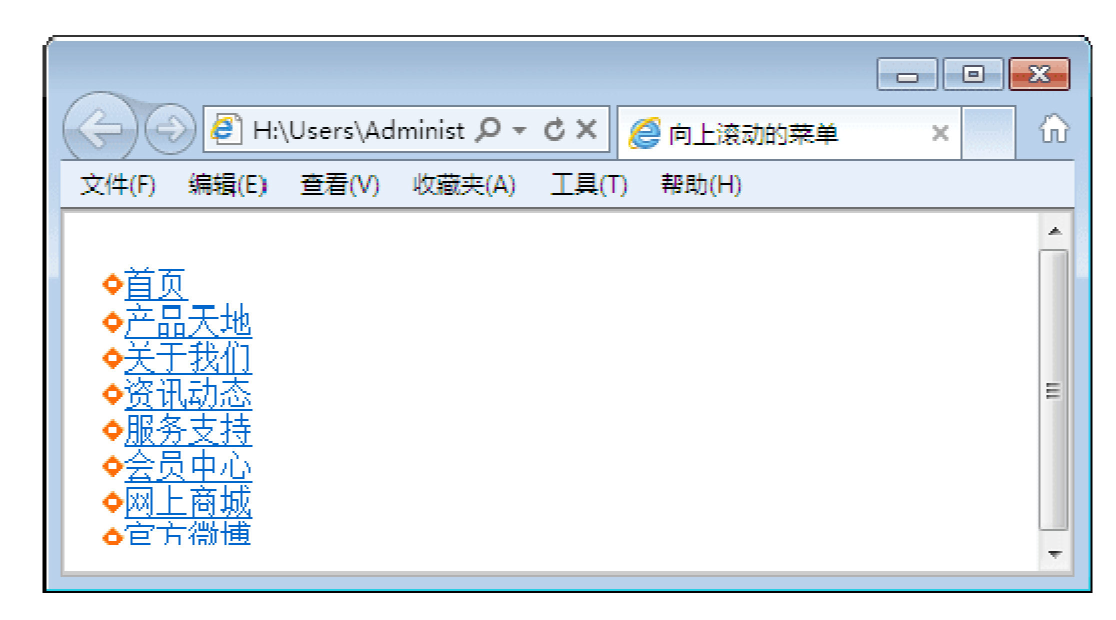


图 20-25　最终效果

20.9　综合实例 7——跟随鼠标移动的图片

在众多网站中，特别是游戏网站或小型商业网站，都喜欢使用鼠标图片跟随的特效，这样一方面可以在鼠标指针旁边加上网站说明的相关信息或者欢迎信息，另一方面也吸引人的注意力，使其更加关注此网站。本实例实现图片跟随鼠标移动的特效，具体步骤如下所示。

01 分析需求。

要实现该效果，需要通过 JavaScript 获取鼠标指针的位置，并且动态调整图片的位置。通过 position 的绝对定位，很容易调整图片位置。采用 CSS 的绝对定位是 JavaScript 调整页面元素时常用的方法。实例完成后，效果如图 20-26 所示。

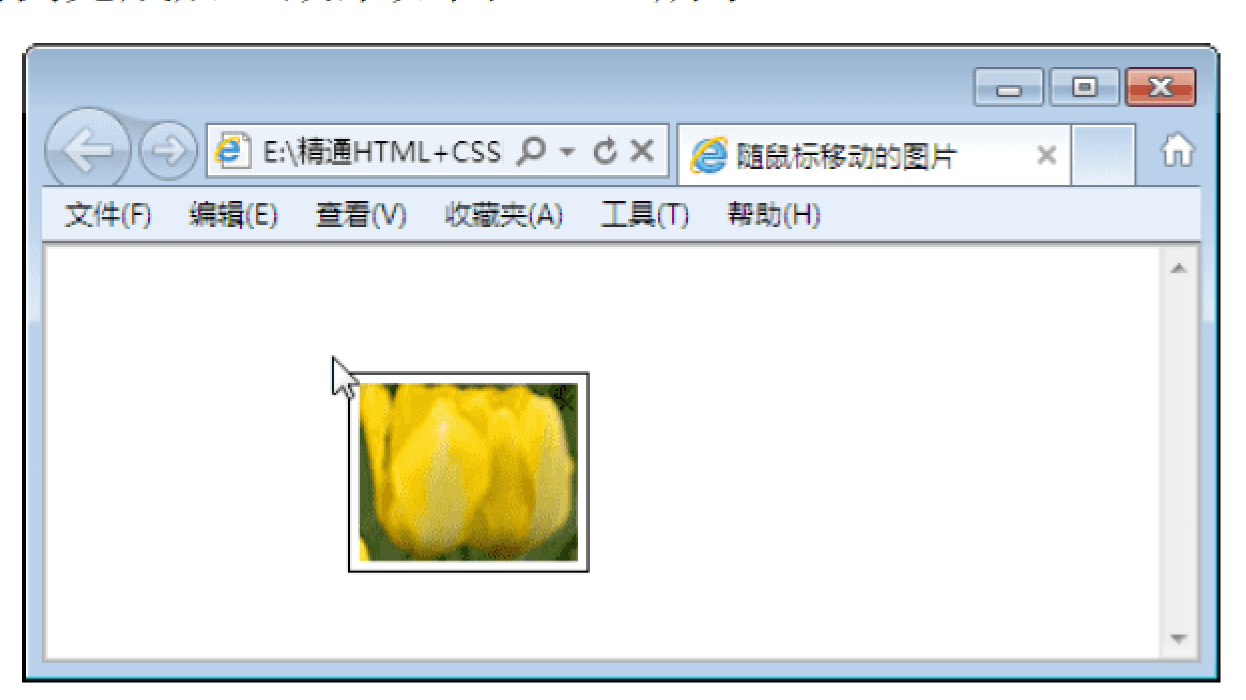


图 20-26　移动的图片

02 创建基本 HTML 页面。



```
<html >
<head>
<title>随鼠标移动的图片</title>
</head>
<body>
</body>
</html>
```

上面的代码比较简单，只实现了一个 HTML 页面结构。

03 添加 JavaScript 代码，实现图片随鼠标移动。

```
<script type="text/javascript">
function badAD(html){
    var ad=document.body.appendChild(document.createElement('div'));
    ad.style.cssText="border:1px solid #000;background:#FFF;position:absolute;padding:4px 4px 4px 4px;
    font: 12px/1.5 verdana;";
    ad.innerHTML=html||'This is bad idea!';
    var c=ad.appendChild(document.createElement('span'));
    c.innerHTML="×";
    c.style.cssText="position:absolute;right:4px;top:2px;cursor:pointer";
    c.onclick=function (){
        document.onmousemove=null;
        this.parentNode.style.left='-99999px'
    };
    document.onmousemove=function (e){
        e=e||window.event;
        var x=e.clientX,y=e.clientY;
        setTimeout(function() {
            if(ad.hover) return;
            ad.style.left=x+5+'px';
            ad.style.top=y+5+'px';
        },120)
    }
    ad.onmouseover=function (){
        this.hover=true
    };
    ad.onmouseout=function (){
        this.hover=false
    }
}
badAD('<img src="18.png">')
</script>
```

在上面的代码中，使用 appendChild()方法为当前页面创建了一个 DIV 对象，并为 DIV 层设置了相应样式。下面的 e.clientX 和 e.clientY 语句确定鼠标位置，并动态调整图片位置，从而实现图片的移动效果。在 IE 9.0 中浏览效果如图 20-27 所示，可以看到鼠标在页面中移动时，图片也跟着移动。

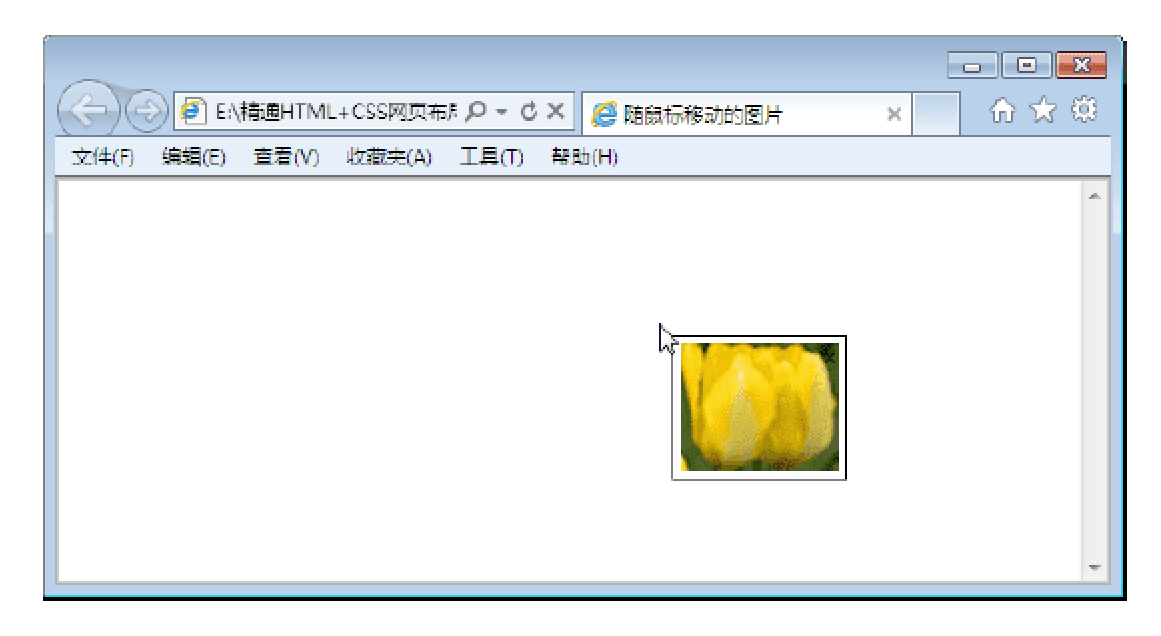


图 20-27　最终效果

20.10　综合实例 8——树形菜单

网站首页的特点之一是需要导航的页面很多，有时不得不将所有需要导航的部分都放到一个导航菜单中。树形导航菜单是网页设计中最常用的菜单之一。本实例将创建一个树形菜单，具体步骤如下所示。

01 分析需求。

要实现一个树形菜单，需要三个方面配合，一个是<ul>无序列表，用于显示的菜单，一个是 CSS 样式，用于修饰树形菜单样式，一个是 JavaScript 程序，用于实现单击鼠标时展开菜单选项。实例完成后，效果如图 20-28 所示。

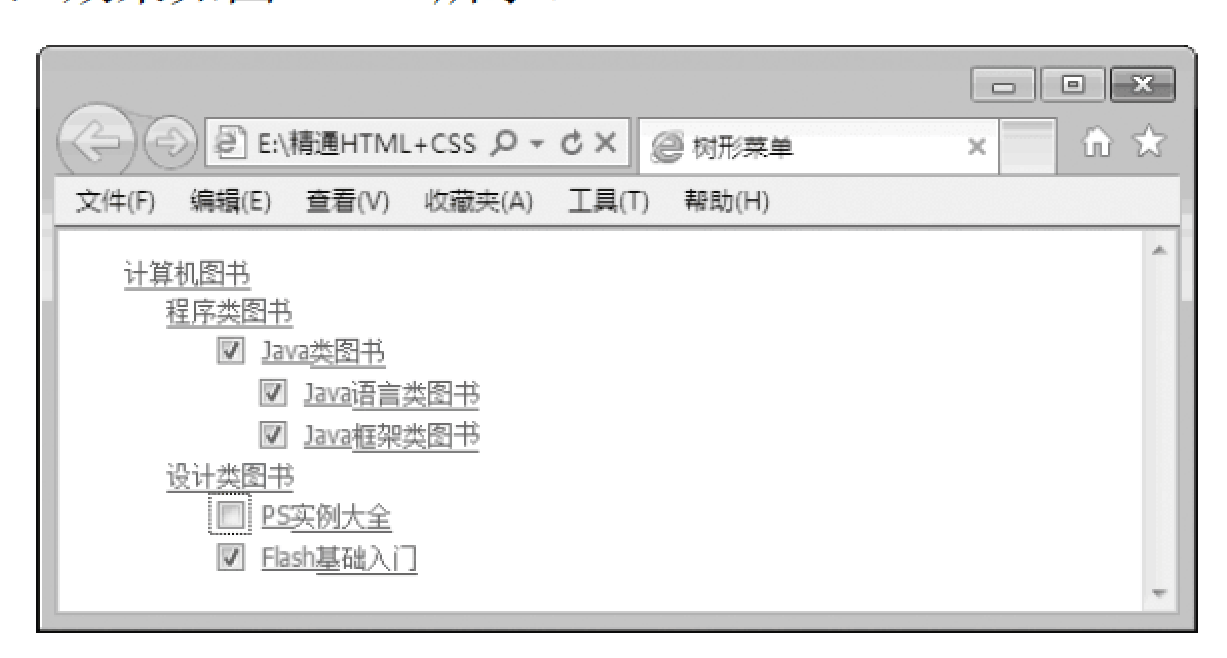


图 20-28　树形菜单

02 创建 HTML 页面，实现菜单列表。

```
<html >
<head>
<title>树形菜单</title>
</head>
<body>
<ul id="menu_zzjs_net">
 <li>
  <label><a href="javascript:;">计算机图书</a></label>
  <ul class="two">
```

```
        <li>
         <label><a href="javascript:;">程序类图书</a></label>
         <ul class="two">
          <li>
           <label><input type="checkbox" value="123456"><a href="javascript:;">Java 类图书</a></label>
           <ul class="two">
            <li><label><input type="checkbox" value="123456"><a href="javascript:;">Java 语言类图书
</a></label></li>
            <li>
             <label><input type="checkbox" value="123456"><a href="javascript:;">Java 框架类图书
</a></label>
             <ul class="two">
              <li>
               <label><input type="checkbox" value="123456"><a href="javascript:;">Struts2 图书</a></label>
               <ul class="two">
                <li><label><input type="checkbox" value="123456"><a href="javascript:;">Struts1</a></label>
</li>
                <li><label><input type="checkbox" value="123456"><a href="javascript:;">Struts2</a></label>
</li>
               </ul>
              </li>
              <li><label><input type="checkbox" value="123456"><a href="javascript:;">Hibernate 入门
</a></label></li>
             </ul>
            </li>
           </ul>
          </li>
         </ul>
        </li>
        <li>
         <label><a href="javascript:;">设计类图书</a></label>
         <ul class="two">
          <li><label><input type="checkbox" value="123456"><a href="javascript:;">PS 实例大全</a></label>
</li>
          <li><label><input type="checkbox" value="123456"><a href="javascript:;">Flash 基础入门</a>
</label></li>
         </ul>
        </li>
       </ul>
      </li>
     </ul>
     </body>
```

```
</html>
```

在 IE 9.0 中浏览效果如图 20-29 所示，可以看到无序列表在页面上显示，并且显示全部元素，字体颜色为蓝色。

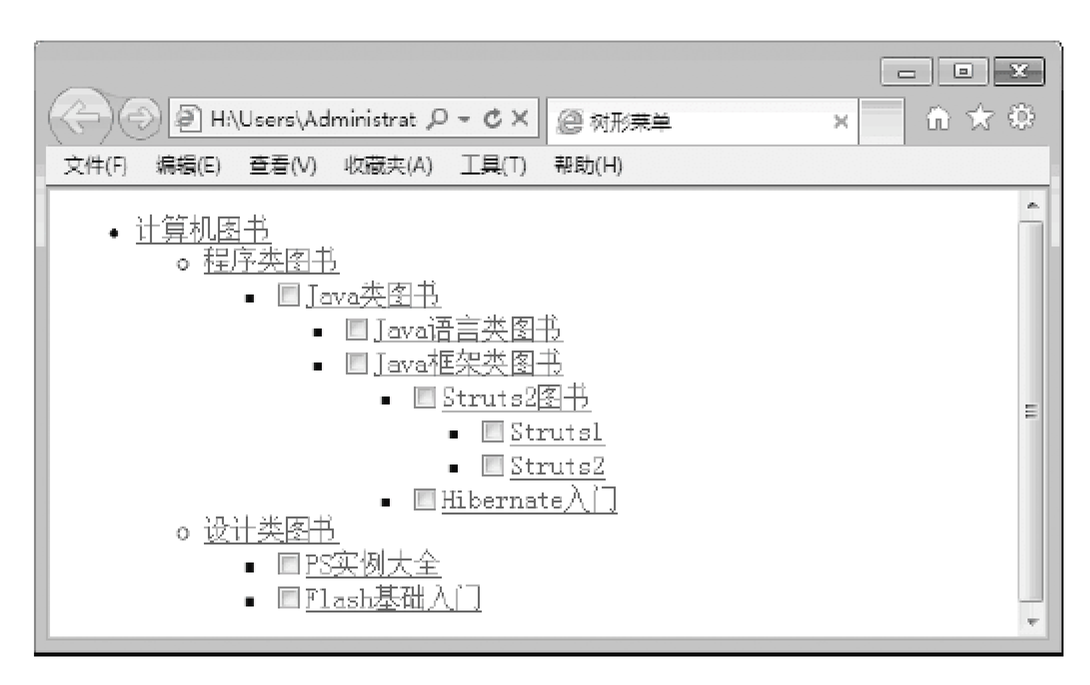


图 20-29　无序列表

03 添加 JavaScript 代码，实现单击发鼠标展开菜单。

```
<script type="text/javascript" >
 function addEvent(el,name,fn){//绑定事件
  if(el.addEventListener) return el.addEventListener(name,fn,false);
  return el.attachEvent('on'+name,fn);
 }
 function nextnode(node){//寻找下一个兄弟并剔除空的文本节点
  if(!node) return ;
  if(node.nodeType == 1)
   return node;
  if(node.nextSibling)
   return nextnode(node.nextSibling);
 }
 function prevnode(node){//寻找上一个兄弟并剔除空的文本节点
  if(!node) return ;
  if(node.nodeType == 1)
   return node;
  if(node.previousSibling)
   return prevnode(node.previousSibling);
 }
 function parcheck(self,checked){//递归寻找父亲元素，并找到 input 元素进行操作
  var par = prevnode(self.parentNode.parentNode.parentNode.previousSibling),parspar;
  if(par&&par.getElementsByTagName('input')[0]){
   par.getElementsByTagName('input')[0].checked = checked;
   parcheck(par.getElementsByTagName('input')[0],sibcheck(par.getElementsByTagName('input')[0]));
  }
 }
```

```
function sibcheck(self){//判断兄弟节点是否已经全部选中
  var sbi = self.parentNode.parentNode.parentNode.childNodes,n=0;
  for(var i=0;i<sbi.length;i++){
    if(sbi[i].nodeType != 1)//由于孩子结点中包括空的文本节点，所以这里累计长度时也要算上
      n++;
    else if(sbi[i].getElementsByTagName('input')[0].checked)
      n++;
  }
  return n==sbi.length?true:false;
}
addEvent(document.getElementById('menu_zzjs_net'),'click',function(e){//绑定 input 点击事件，使用
menu_zzjs_net 根元素代理
  e = e||window.event;
  var target = e.target||e.srcElement;
  var tp = nextnode(target.parentNode.nextSibling);
  switch(target.nodeName){
    case 'A'://点击 A 标签展开和收缩树形目录，并改变其样式
      if(tp&&tp.nodeName == 'UL'){
        if(tp.style.display != 'block' ){
          tp.style.display = 'block';
          prevnode(target.parentNode.previousSibling).className = 'ren'
        }else{
          tp.style.display = 'none';
          prevnode(target.parentNode.previousSibling).className = 'add'
        }
      }
      break;
    case 'SPAN'://点击图标只展开或者收缩树形目录
      var ap = nextnode(nextnode(target.nextSibling).nextSibling);
      if(ap.style.display != 'block' ){
        ap.style.display = 'block';
        target.className = 'ren'
      }else{
        ap.style.display = 'none';
        target.className = 'add'
      }
      break;
    case 'INPUT'://点击 checkbox，选中父亲元素，则孩子节点中的 checkbox 也同时被选中，孩子结点取
    消父元素也随之取消
      if(target.checked){
        if(tp){
          var checkbox = tp.getElementsByTagName('input');
```

```
            for(var i=0;i<checkbox.length;i++)
              checkbox[i].checked = true;
          }
        }else{
          if(tp){
            var checkbox = tp.getElementsByTagName('input');
            for(var i=0;i<checkbox.length;i++)
              checkbox[i].checked = false;
          }
        }
        parcheck(target,sibcheck(target));//当孩子结点取消选中的时候，调用该方法递归其父节点的 checkbox
逐一取消选中
        break;
    }
  });
  window.onload = function(){//页面加载时给有孩子结点的元素动态添加图标
    var labels = document.getElementById('menu_zzjs_net').getElementsByTagName('label');
    for(var i=0;i<labels.length;i++){
      var span = document.createElement('span');
      span.style.cssText ='display:inline-block;height:18px;vertical-align:middle;width:16px;cursor:pointer;';
      span.innerHTML = ' '
      span.className = 'add';
      if(nextnode(labels[i].nextSibling)&&nextnode(labels[i].nextSibling).nodeName == 'UL')
        labels[i].parentNode.insertBefore(span,labels[i]);
      else
        labels[i].className = 'rem'
    }
  }
</script>
```

在 IE 9.0 中浏览效果如图 20-30 所示，可以看到无序列表在页面上显示，使用鼠标单击可以展开或关闭相应的选项，但其样式非常难看。

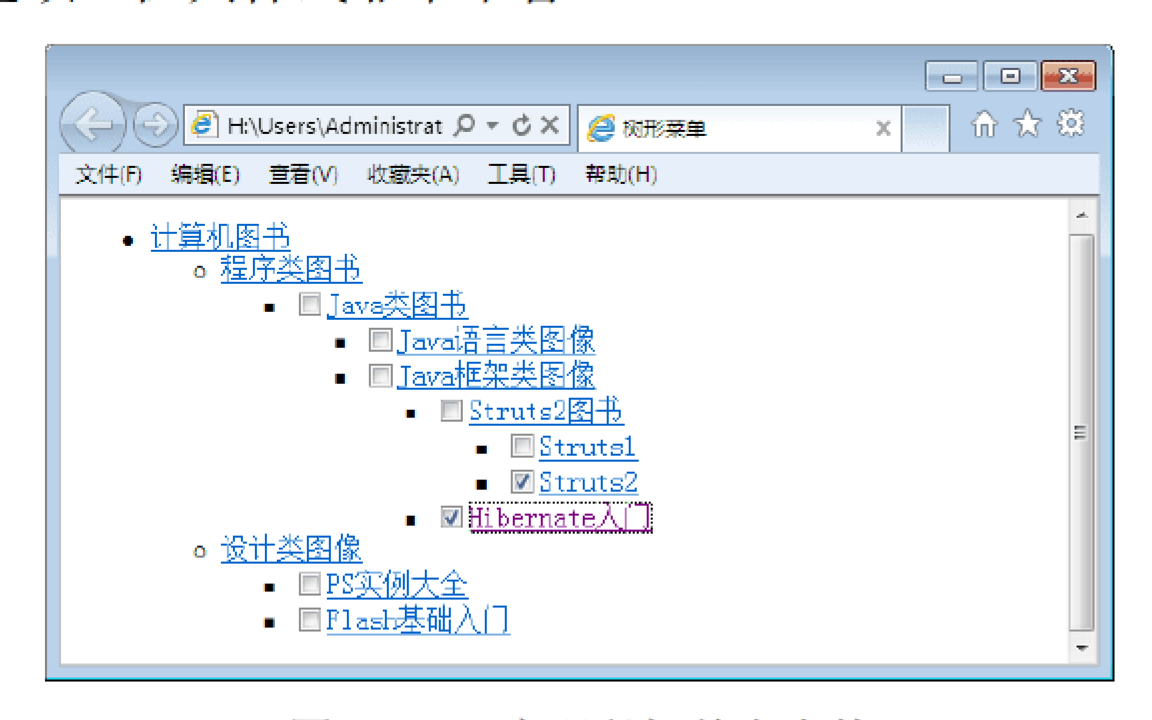


图 20-30 实现鼠标单击事件

04 添加 CSS 代码，修饰列表选项。

```
<style type="text/css">
body{margin:0;padding:0;font:12px/1.5 Tahoma,Helvetica,Arial,sans-serif;}
ul,li,{margin:0;padding:0;}
ul{list-style:none;}
#menu_zzjs_net{margin:10px;width:200px;overflow:hidden;}
#menu_zzjs_net li{line-height:25px;}
#menu_zzjs_net .rem{padding-left:16px;}
#menu_zzjs_net .add{background:url() -4px -31px no-repeat;}
#menu_zzjs_net .ren{background:url() -4px -7px no-repeat;}
#menu_zzjs_net li a{color:#666666;padding-left:5px;outline:none;blr:expression(this.onFocus=this.blur());}
#menu_zzjs_net li input{vertical-align:middle;margin-left:5px;}
#menu_zzjs_net .two{padding-left:20px;display:none;}
</style>
```

在 IE 9.0 中浏览效果如图 20-31 所示，可以看到与原来的页面相比，菜单样式更加漂亮。

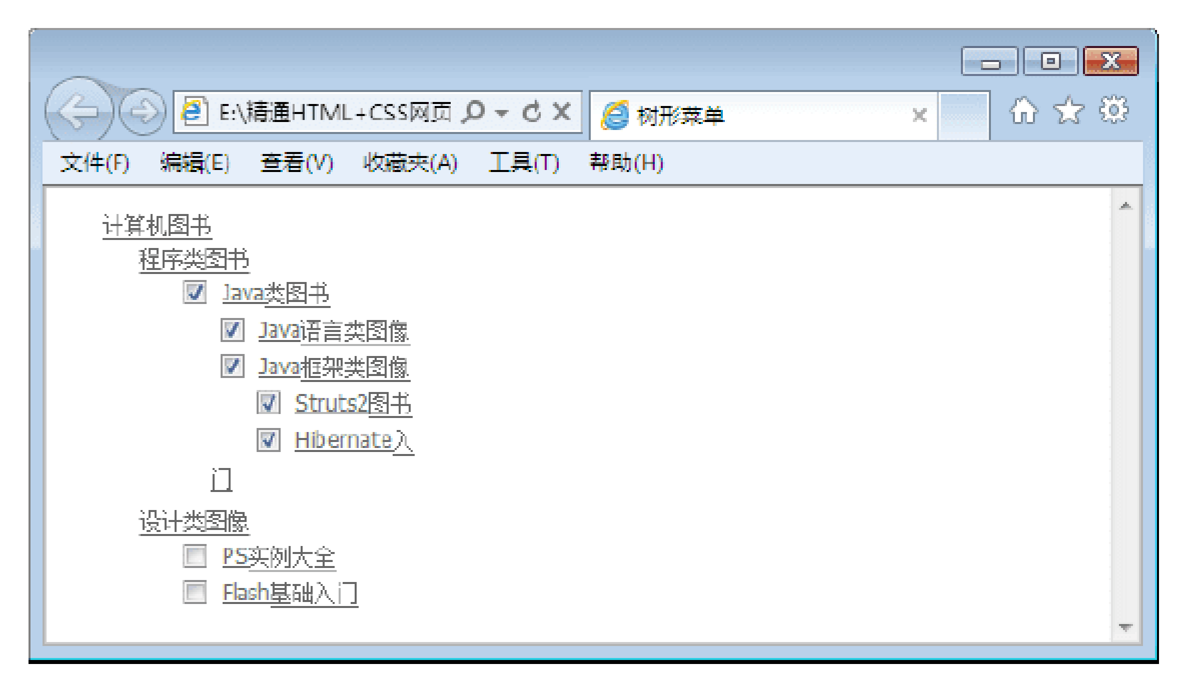


图 20-31 最终效果

20.11 综合实例 9——颜色选择器

在页面中定义背景色和字体颜色是常见的一种操作，而在选取颜色时会比较发愁，不知道哪种颜色适合，并且也不知道颜色值。此时可以利用颜色选择器来定义颜色并获取颜色值。本实例将创建一个颜色选择器，具体步骤如下所示。

01 分析需求。

本实例原理非常简单，就是将几个常用的的颜色值进行组合，组合在一起后合并，就是所要选择的颜色值。这些都是利用 JS 代码完成的。实例完成后，效果如图 20-32 所示。

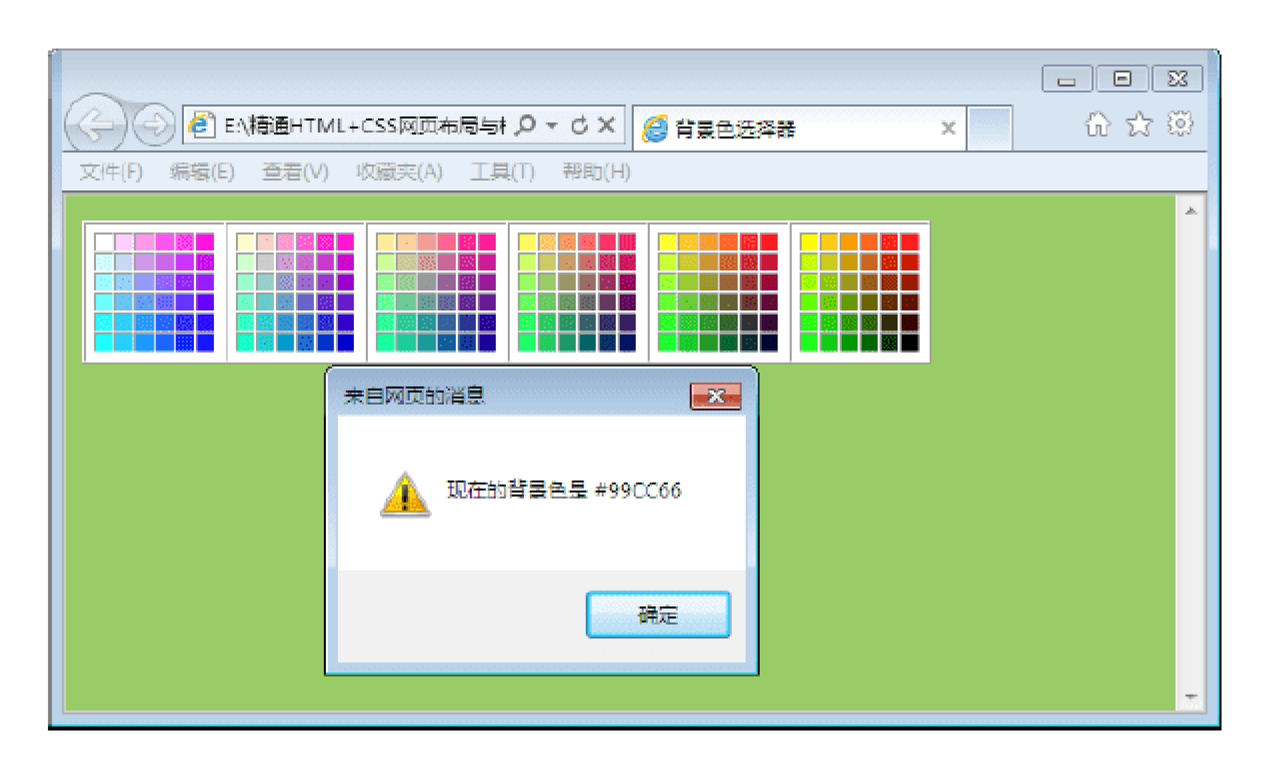


图 20-32　设定页面背景色

02 创建基本 HTML 页面。

```
<html>
<head><title>背景色选择器</title>
</head>
<body bgcolor="#FFFFFF">
</body></html>
```

上述代码比较简单，只实现了一个页面框架。

03 添加 JavaScript 代码，实现颜色选择。

```
<script language="JavaScript">
<!--
var hex = new Array(6)
hex[0] = "FF"
hex[1] = "CC"
hex[2] = "99"
hex[3] = "66"
hex[4] = "33"
hex[5] = "00"
function display(triplet)
{
   document.bgColor = '#' + triplet
   alert('现在的背景色是  #'+triplet)
}
function drawCell(red, green, blue)
{
   document.write('<TD BGCOLOR="#' + red + green + blue + '">')
   document.write('<A HREF="javascript:display(\'' + (red + green + blue) + '\')">')
   document.write('<IMG SRC="place.gif" BORDER=0 HEIGHT=12 WIDTH=12>')
   document.write('</A>')
```

```
document.write('</TD>')
}
function drawRow(red, blue)
{
  document.write('<TR>')
  for (var i = 0; i < 6; ++i)
  {
    drawCell(red, hex[i], blue)
  }   document.write('</TR>')
}function drawTable(blue)
{
  document.write('<TABLE CELLPADDING=0 CELLSPACING=0 BORDER=0>')
  for (var i = 0; i < 6; ++i)
  {
    drawRow(hex[i], blue)
  }
  document.write('</TABLE>')
}
function drawCube()
{
  document.write('<TABLE CELLPADDING=5 CELLSPACING=0 BORDER=1><TR>')
  for (var i = 0; i < 6; ++i)
  {
    document.write('<TD BGCOLOR="#FFFFFF">')
    drawTable(hex[i])
    document.write('</TD>')
  }   document.write('</TR></TABLE>')
}drawCube()
// --></script>
```

在上面的代码中，创建了一个数组对象 hex 用来存放不同的颜色值。下面几个函数分别将数组中颜色组合在一起，并在页面显示，display 函数完成背景颜色的定义并显示颜色值。

在 IE 9.0 中浏览效果如图 20-33 所示，可以看到页面中显示多个表格，每个单元格代表一种颜色。

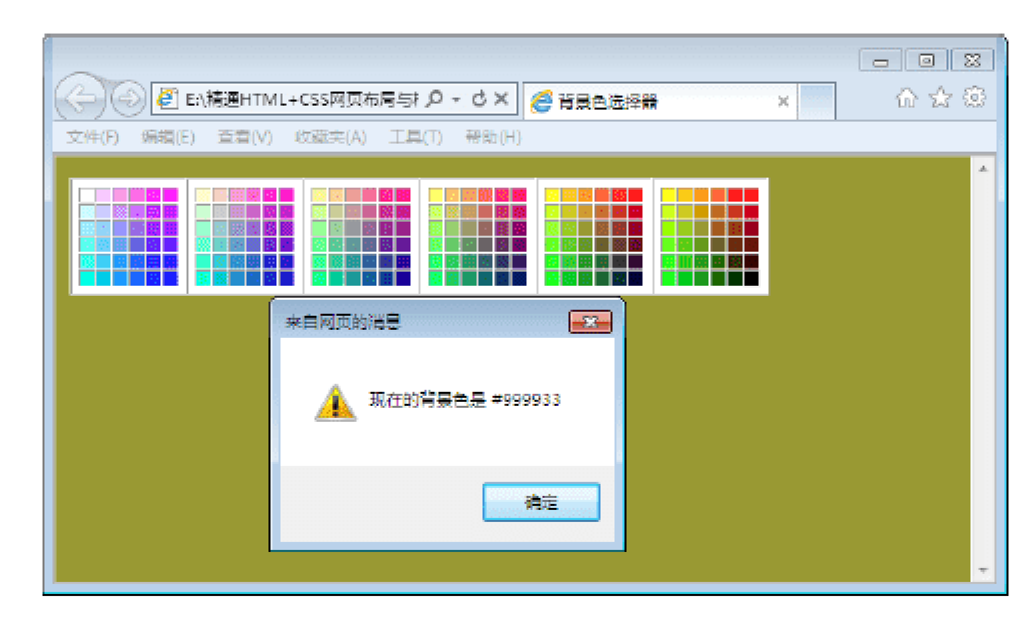


图 20-33　最终效果

20.12　专家解惑

1. JavaScript 的数组中常用的方法有哪些？

- jion()　将 array 中的所有 element 以 string 的形式连在一起

```
var a = [1,2,3];
s = a.join();          // s == "1,2,3"
s = a.join(": ");   // s == "1: 2: 3"
```

- reverse()　将数组中的元素顺序颠倒

```
var a = [1,2,3];
a.reverse();
s = a.join(); // s == "3,2,1"
```

- sort()　排序，默认按字母顺序排序，区分大小写，可以自定义排序方式.

```
var a = [111,4,33,2];
a.sort();   // a == [111,2,33,4]
a.sort(function(a,b) { // a == [2,4,33,111]
        return a-b;           // Returns < 0, 0, or > 0
   });
```

- concat()　连接多个数组

```
var a = [1,2,3];
a.concat(4,5);                 // return [1,2,3,4,5]
a.concat([4,5]);              // return [1,2,3,4,5]
a.concat([4,5], [6,7])   // return [1,2,3,4,5,6,7]
a.concat(4,[5,[6,7]]);   // return [1,2,3,4,5,6,7]
```

2. JavaScript 中的注释有哪些？

可以在程序中添加注释来描述代码的功能，通过注释还可以使一些代码无效，以便逐行检查程序，及时发现并解决问题。JavaScript 主要提供了 3 种注释的方法：

- 单行注释：在需要注释的代码前添加字符"//"，//后面的部分会被注释。
- 多行注释：在需要注释代码前添加"/*"，之后添加"*/"，两个符号之间的部分会被注释。
- HTML 注释：使用传统的 HTML 注释，<!--和-->之间的部分会被注释，注释内容可以为一行或多行。

第 21 章

CSS与XML的综合运用

XML 是一种标准化的文本格式，可以在 Web 上表示结构化信息，利用它可以存储有复杂结构的数据信息。XML 是 HTML 的补充，但 XML 并不是 HTML 的替代品。在将来的网页开发中，XML 将被用来描述和存储数据，而 HTML 则用来格式化和显示数据。

21.1 XML 语法基础

XML 是标记语言，支持开发者为 Web 信息设计自己的标记。XML 要比 HTML 强大得多，它不再是固定的标记，而是允许定义数量不限的标记来描述文档中的资料，允许嵌套的信息结构。

21.1.1 XML 概念

随着因特网的发展，为了控制网页显示样式，增加了一些描述如何显示数据的标记，如<center>、<b>等标记。但随着 HTML 的不断发展，W3C 组织逐渐意识到 HTML 存在一些无法避免的问题。

- 不能解决所有解释数据的问题，如影音文件或化学公式、音乐符号等其他形式的内容。
- 效能问题，需要下载整份文件才能开始对文件做搜寻的动作。
- 扩充性、弹性、易读性均不佳。

为了解决以上问题，专家们依照 HTML 的发展规律，制定出一套规则严谨但是语法简单的描述数据语言：XML。

XML（eXtensible Markup Language，可扩展标记语言）是 W3C 推荐的参考通用标记语言，同样也是 SGML 的子类，可以定义自己的一组标记。它具有下面几个特点。

- XML 是一种元标记语言，所谓“元标记语言”就是开发者可以根据需要定义自己的标记，例如开发者可以定义标记<book>、<name>，任何满足 XML 命名规则的名称都可以作为标记，这就为不同应用程序的应用打开了大门。
- 允许通过使用自定义格式，标识、交换和处理数据库可以理解的数据。
- 基于文本的格式，允许开发人员描述结构化数据并在各种应用之间发送和交换这些数据。

- 有助于在服务器之间传输结构化数据。
- XML 使用的是非专有的格式，不受版权、专利、商业秘密或是其他类型的知识产权的限制。

XML 的功能是非常强大的，同时对于人类或是计算机程序来说，都容易阅读和编写。因而 XML 成为交换语言的首选。网络的最大好处是信息共享，在不同的计算机中发送数据，而 XML 用来告诉我们“数据是什么”，利用 XML 可以在网络上交换任何一种信息。

【例 21.1】（实例文件：ch21\21.1.xml）

```
<?xml version="1.0" encoding="GB2312" ?>
<电器>
    <家用电器>
        <品牌>小天鹅洗衣机</品牌>
        <购买时间>2009-10-01</购买时间>
        <价格  币种="人民币">899 元</价格>
    </家用电器>
    <家用电器>
        <品牌>海尔冰箱</品牌>
        <购买时间>2011-08-16</购买时间>
        <价格  币种="人民币">3990</价格>
    </家用电器>
</电器>
```

此处需要将文件保存为 XML 文件。该文件中，每个标记都是用汉语编写的，是自定义标记。可以将电器看做一个对象，该对象包含了多个家用电器，家用电器是用来存储电器的相关信息的，也可以说家用电器对象是一种数据结构模型。在页面中没有对数据的样式进行修饰，而只告诉我们数据结构是什么，数据是什么。

在 IE 9.0 中浏览效果如图 21-1 所示，可以看到整个页面以树形结构显示，通过单击“-”可以关闭整个树形结构，单击“+”可以展开树形结构。

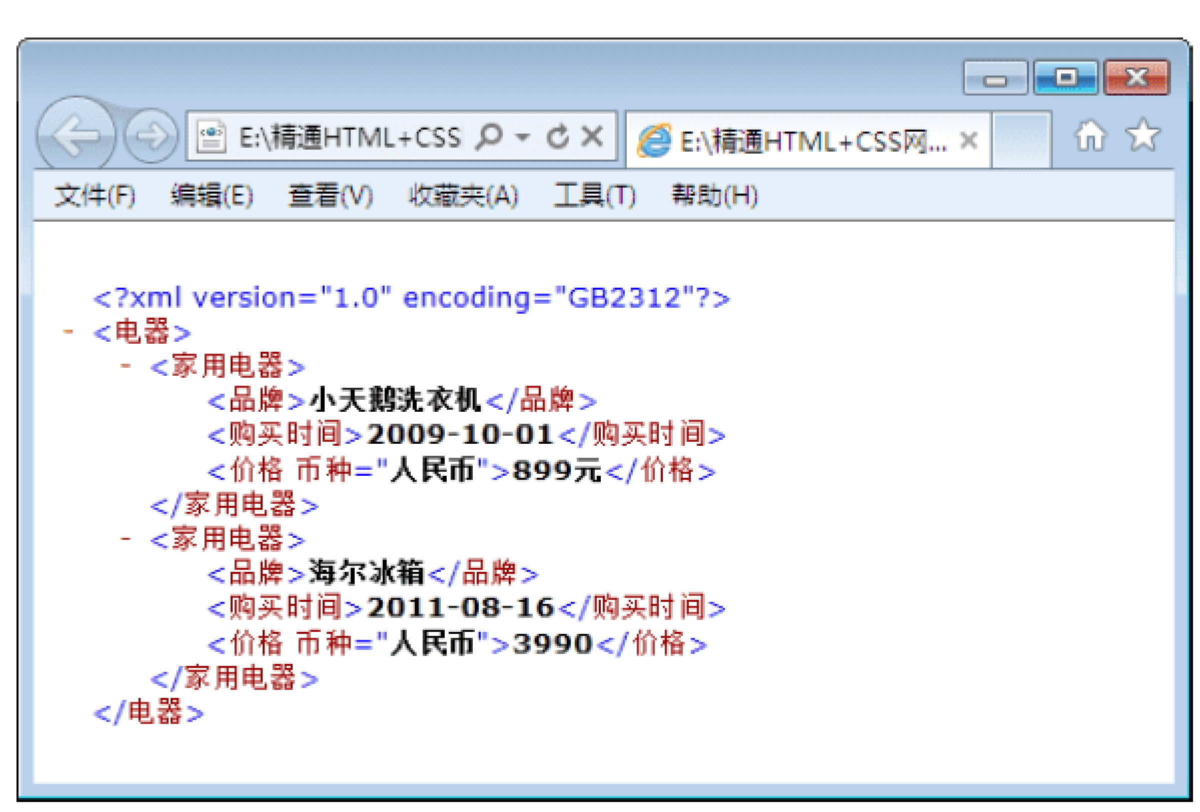


图 21-1　XML 文件显示

21.1.2　XML 文档的组成和声明

一个完整的 XML 文档由声明、元素、注释、字符引用和处理指令组成。在文档中，所有这些 XML 文档的组成部分都是通过元素标记来指明的。可以将 XML 文档分为 3 个部分，如图 21-2 所示。

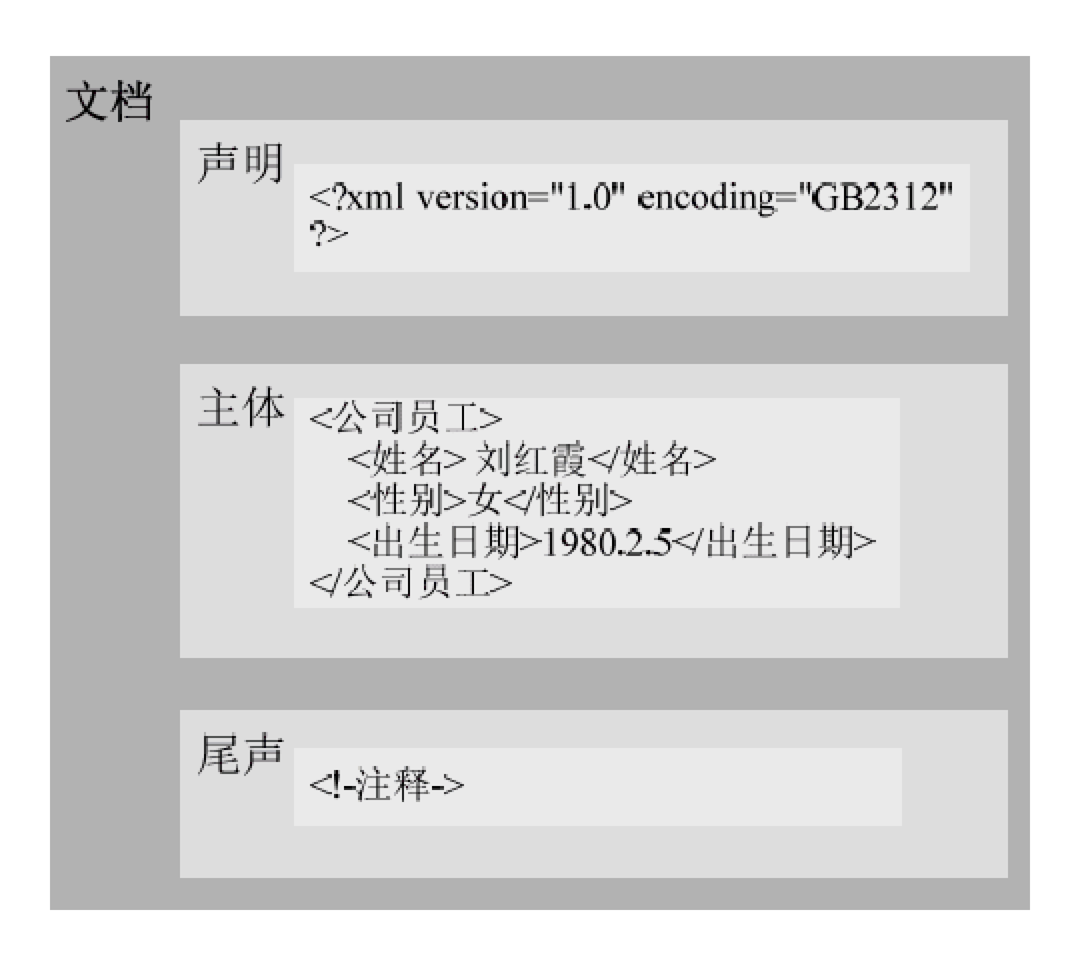


图 21-2　XML 文档组成

XML 声明必须作为 XML 文档的第一行，前面不能有空白、注释或其他处理指令。完整的声明格式如下：

```
<?xml version="1.0" encoding="编码" standalone="yes/no" ?>
```

其中 version 属性不能省略，且必须排在属性列表中的第一位，指明所采用的 XML 版本号，值为 1.0。该属性用来保证对 XML 未来版本的支持。encoding 属性是可选属性。该属性指定了文档采用的编码方式，即规定了采用哪种字符集对 XML 文档进行字符编码，常用的编码方式为 UTF-8 和 GB2312。如果没有使用 encoding 属性，那么该属性的默认值是 UTF-8，如果 encoding 属性值设置为 GB2312，则文档必须使用 ANSI 编码保存，文档的标记以及标记内容只能使用 ASCII 字符和中文。

使用 GB2312 编码的 XML 声明如下：

```
<?xml version="1.0" encoding="GB2312" ?>
```

XML 文档主体必须有根元素。所有的 XML 必须包含可定义根元素的单一标记对。所有其他的元素都必须处于这个根元素内部。所有的元素均可拥有子元素。子元素必须被正确地嵌套于它们的父元素内部。根标记以及根标记内容共同构成 XML 文档主体。没有文档主体的 XML 文档将不会被浏览器或其他 XML 处理程序所识别。

注释可以提高文档的阅读性，尽管 XML 解析器通常会忽略文档中的注释，但位置适当且有意义的注释可以大大提高文档的可读性。所以 XML 文档中不用于描述数据的内容都可以包含在注释中，注释以“<!--”开始，以“-->”结束，在起始符和结束符之间为注释内容，注释内容中可以输入符合注释规则的任何字符串。

【例 21.2】（实例文件：ch21\21.2.xml）

```
<?xml version="1.0" encoding="gb2312"?>
<!--这是一个优秀学生名单-->
<学生名单>
<学生>
  <姓名>张三</姓名>
    <学号>21</学号>
    <性别>男</性别>
</学生>
<学生>
  <姓名>李四</姓名>
    <学号>22</学号>
      <性别>女</性别>
</学生>
</学生名单>
```

在上面的代码中，第一句代码是一个 XML 声明。“<学生>”标记是“<学生名单>”标记的子元素，而“<姓名>”标记和“<学号>”标记是“<学生>”标记的子元素。“<!—这是一个优秀学生名单-->”是一个注释。

在 IE 9.0 中浏览效果如图 21-3 所示，可以看到页面中显示了一个树形结构，并且数据层次感非常好。

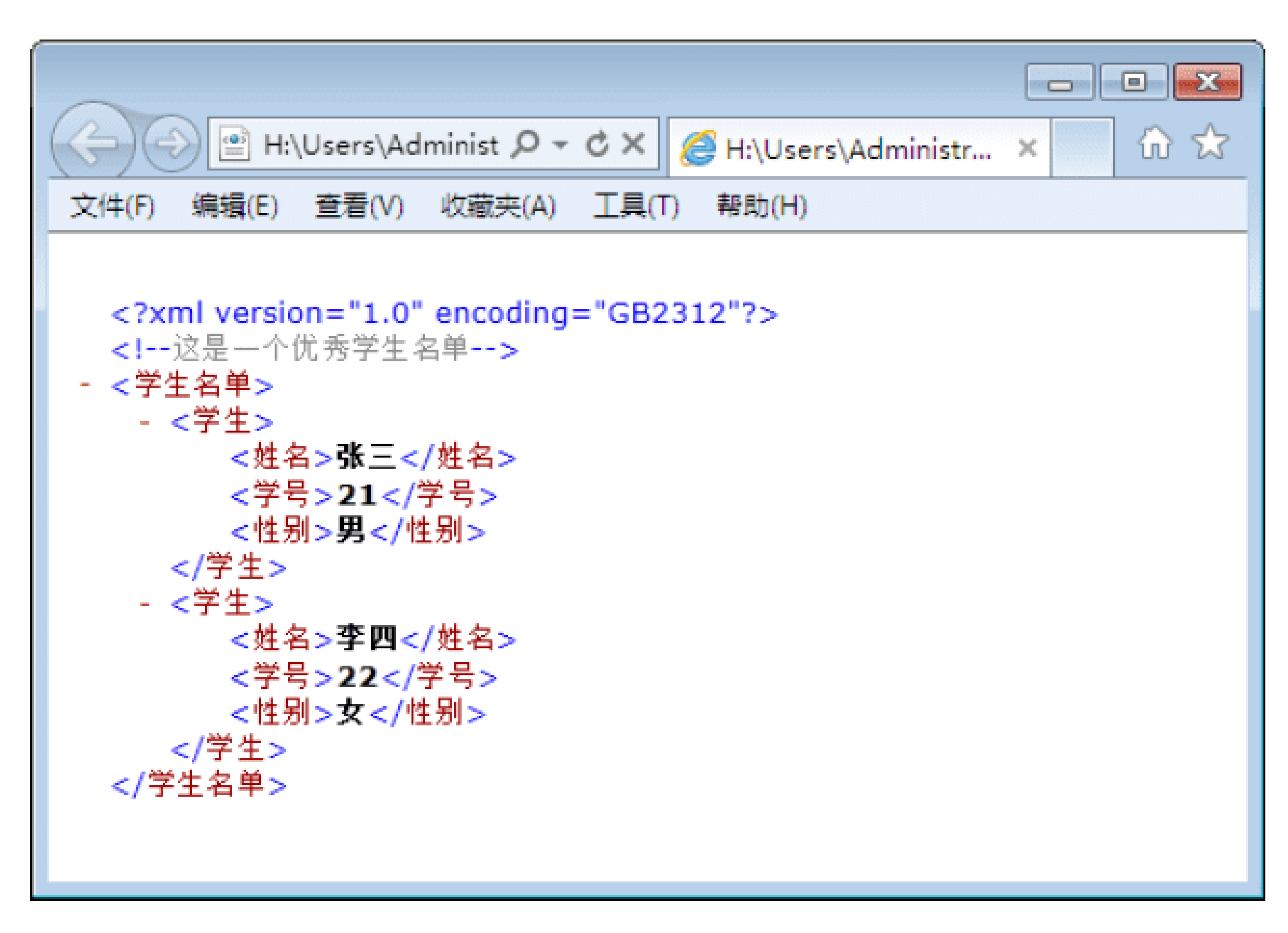


图 21-3　XML 文档的组成

21.1.3　XML 元素介绍

元素是以树形分层结构排列的，它可以嵌套在其他元素中。

1．元素类别

在 XML 文档中，元素也分为非空元素和空元素两种类型。一个 XML 非空元素是由开始

标记、结束标记以及标记之间的数据构成的。开始标记和结束标记用来描述标记之间的数据。标记之间的数据被认为是元素的值。非空元素的语法结构如下所示：

```
<开始标记>文本内容</结束标记>
```

而空元素就是不包含任何内容的元素，即开始标记和结束标记之间没有任何内容。其语法结构如下所示：

```
<开始标记></结束标记>
```

可以把元素内容为文本的非空元素转换为空元素。例如，

```
<hello>下午好</hello>
```

<hello>是一个非空元素，如果把非空元素的文本内容转换为空元素的属性，那么转换后的空元素可以写为。

```
<hello content="下午好"></hello>
```

2．元素命名规范

XML 元素命名规则与 Java、C 等命名规则类似，它也是一种对大小写敏感的语言。XML 元素的命名必须遵守下列规则：

- 元素名中可以包含字母、数字和其他字符。

如<place>、<地点>、<no123>等。元素名中虽然可以包含中文，但是在不支持中文的环境中将不能解释包含中文字符的 XML 文档。

- 元素名中不能以数字或标点符号开头，如<123no>、<.name>、<?error>元素名称都是非法名称。
- 元素名中不能包含空格，如<no 123>。

3．元素嵌套

元素的内容可以包含子元素。子元素本身也是元素，被嵌套在上层元素之内。如果子元素嵌套了其他元素，那么它同时也是父元素，例如下面的代码所示：

```
<?xml version="1.0" encoding="gb2312" ?>
<students>
  <student>
    <name>张三</name>
    <age>20</age>
  </student>
  ...
</students>
```

<student>是<students>的子元素，同时也是<name>和<age>的父元素，而<name>和<age>是<student>的子元素。

4．元素实例

【例 21.3】（实例文件：ch21\21.3.xml）

```
<?xml version="1.0" encoding="gb2312" ?>
<通讯录>
  <!--"记录"标记中包含姓名、地址、电话和电子邮件 -->
  <记录 date="2011/2/1">
    <姓名>张三</姓名>
    <地址>中州大道 1 号</地址>
    <电话>0371-12345678</电话>
    <电子邮件>rose@tom.com</电子邮件>
  </记录>
  <记录 date="2011/3/12">
    <姓名>李四</姓名>
    <地址>邯郸市工农大道 2 号</地址>
    <电话>123456</电话>
  </记录>
  <记录 date="2011/6/23">
    <姓名>闫阳</姓名>
    <地址>长春市幸福路 6 号</地址>
    <电话>0431-123456</电话>
    <电子邮件>yy@sina.com</电子邮件>
  </记录>
</通讯录>
```

在上面的代码中，第一行是 XML 声明，它声明该文档是 XML 文档，并声明文档所遵守的版本号以及文档使用的字符编码集。在这个例子中，遵守的是 XML 1.0 版本规范，字符编码是 GB2312 编码方式。<记录>是<通讯录>的子标记，但<记录>标记同时是<姓名>和<地址>等标记的父元素。

在 IE 9.0 中浏览效果如图 21-4 所示，可以看到页面中显示了一个树形结构，每个标记中间包含相应的数据。

图 21-4　元素包含

21.2　使用 CSS 修饰 XML 文件

我们知道 XML 文档本身只包含数据，而没有关于显示数据样式的信息。如果需要将 XML 文档数据美观的显示出来，而不是以树形结构显示，可以通过 CSS 来控制 XML 文档中各个元素的呈现方式。

21.2.1　在 XML 文件中使用 CSS

XML 文档数据需要使用 CSS 属性定义显示样式，其方法是把 CSS 代码做成独立的文件，然后引入到 XML 中。在 XML 文档中引入样式表 CSS，可以将数据的内容和表示分离出来，并且能够实现 CSS 的重复使用。

要在 XML 文件中引用 CSS 文件，必须使用下面的操作指令：

```
<?xml-stylesheet href="URI" type="text/css"?>
```

xml-stylesheet 表示在这里使用了样式表。样式表的 URI 表示要引入文件所在的路径，如果只是一个文件的名字，该 CSS 文件必须和 XML 文档同在一个目录下；如果 URI 是一个链接，该链接必须是有效的、可访问的，type 表示该文件所属的类型是文本形式的，其内容是 CSS 代码。

【例 21.4】（实例文件：ch21\21.4.xml）

```
<?xml version="1.0" encoding="GB2312" ?>
<?xml-stylesheet type="text/css" href="20.4.css"?>
<student>
<name>孙福全</name>
<sex>男</sex>
<name>王小玲</name>
<sex>女</sex>
```

```
</student>
```

【例 21.5】（实例文件：ch21\21.4.css）

```
student{
background-color: #ddeecc;
font-family:"幼圆";
text-align:center;
display:block;
}
name{
font-size:20px;
color:red;
}
sex{
font-size:12px;
font-style:italic;
}
```

在 CSS 文件中，针对 student、name 和 sex 3 个标记，设置了不同的显示样式，如字体大小、字体颜色、对齐方式等。

在 IE 9.0 中浏览效果如图 21-5 所示，可以看到 XML 文档不再以树形结构显示，并且没有标记出现，只是显示其标记中数据。

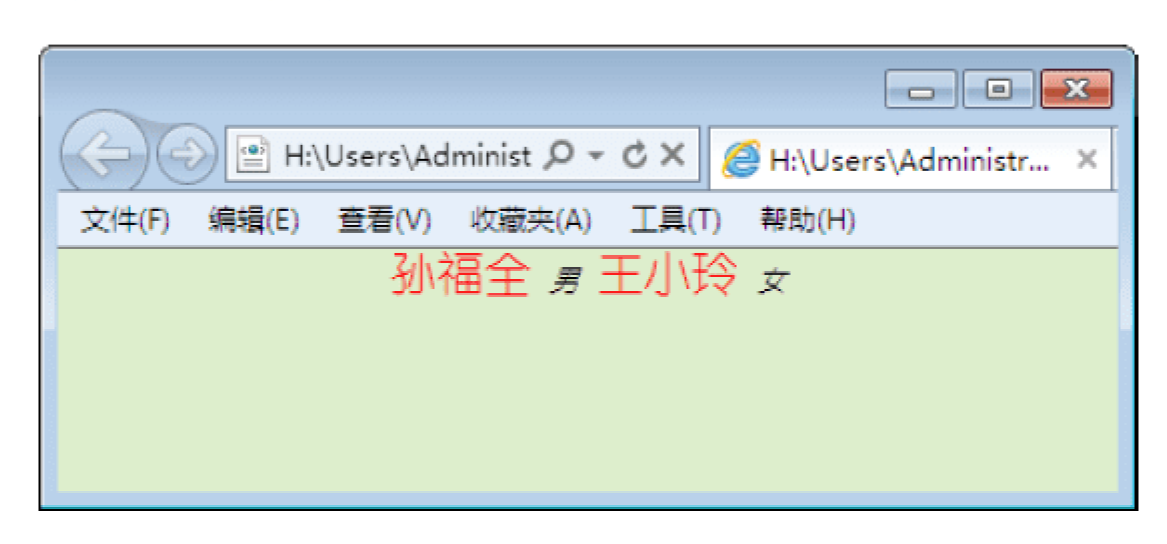


图 21-5 在 XML 文件中引入 CSS 文件

21.2.2 设置字体属性

CSS 样式表则提供了多种字体属性，便于设计者对字进行详细的设置，从而丰富页面效果。例如 font-style、font-variant、font-weight、font-size 和 font-family 等属性，这些属性前面已经介绍过，这里不再赘述。这些字体属性也可以应用于 XML 文件元素。

【例 21.6】（实例文件：ch21\21.5.xml）

```
<?xml version="1.0" encoding="gb2312"?>
<?xml-stylesheet href="20.5.css" type="text/css"?>
<company>
  <name>水月网页设计工作室</name>
```

```
    <address>郑州市花园路松风大夏</address>
    <phone>13012345678</phone>
</company>
```

【例 21.7】（实例文件：ch21\21.5.css）

```
company{
    color: #ddeecc;
    font:normal small-caps bolder 15pt "幼圆" ;
    background-color:#123543
}
name{
     font-size:30px;
     display:block;
}
address{
     font-size: 12px;
     display:block;
}
phone{
     font-size: 12px;
     font-style:italic;
     display:block;
}
```

上面的 CSS 代码，针对 XML 中的标记进行了字体设置、背景颜色和前景色设置。

在 IE 9.0 中浏览效果如图 21-6 所示，可以看到网页中显示了一个公司介绍信息，其中字体大小不一样，联系方式以斜体显示。

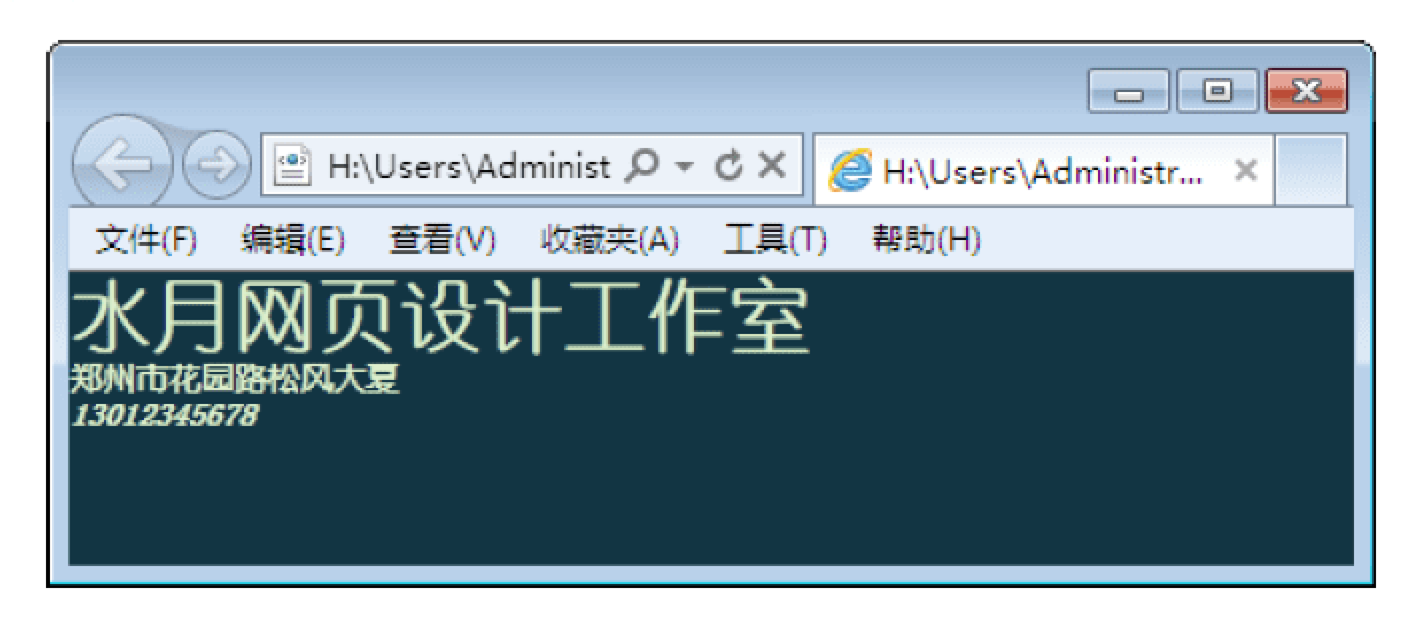


图 21-6　使用 CSS 定义 XML 字体属性

21.2.3　设置色彩属性

颜色和背景是网页设计时两个重要的因素，一个颜色搭配协调、背景优美的文档总能吸引不少的访问者。CSS 的强大表现功能在控制颜色和背景方面同样发挥得淋漓尽致。XML 元素的背景可设置成一种颜色或一幅影像。

在 CSS 中，如果需要设置文本颜色，即网页前景色，通常使用 color 属性，定义元素背景时，其相关属性分别如下：background-color、background-image、background-repeat、background-attachment、background-position。这些前面都已经介绍过，这里不再赘述。

【例 21.8】（实例文件：ch21\21.6.xml）

```
<?xml version="1.0" encoding="GB2312" ?>
<?xml-stylesheet href="20.6.css" type="text/css" ?>
<img>
插花
</img>
```

【例 21.9】（实例文件：ch21\21.6.css）

```
img{
  display:block;
  color:red;
  text-align:center;
  font-size:40px;
  left:50px;
  top:170px;
  background-image:URL("08.jpg");
  background-repeat:no-repeat;
  background-position:left;
}
```

上面的 CSS 代码设置背景以块显示，字体颜色为红色，字体大小为 40 像素，并居中显示。background-image 引入背景图片为 08.jpg，并设置了图片不重复。

在 IE 9.0 中浏览效果如图 21-7 所示，可以看到页面背景为一张图片，且不重复，在图片上显示了“插花”，颜色为红色。

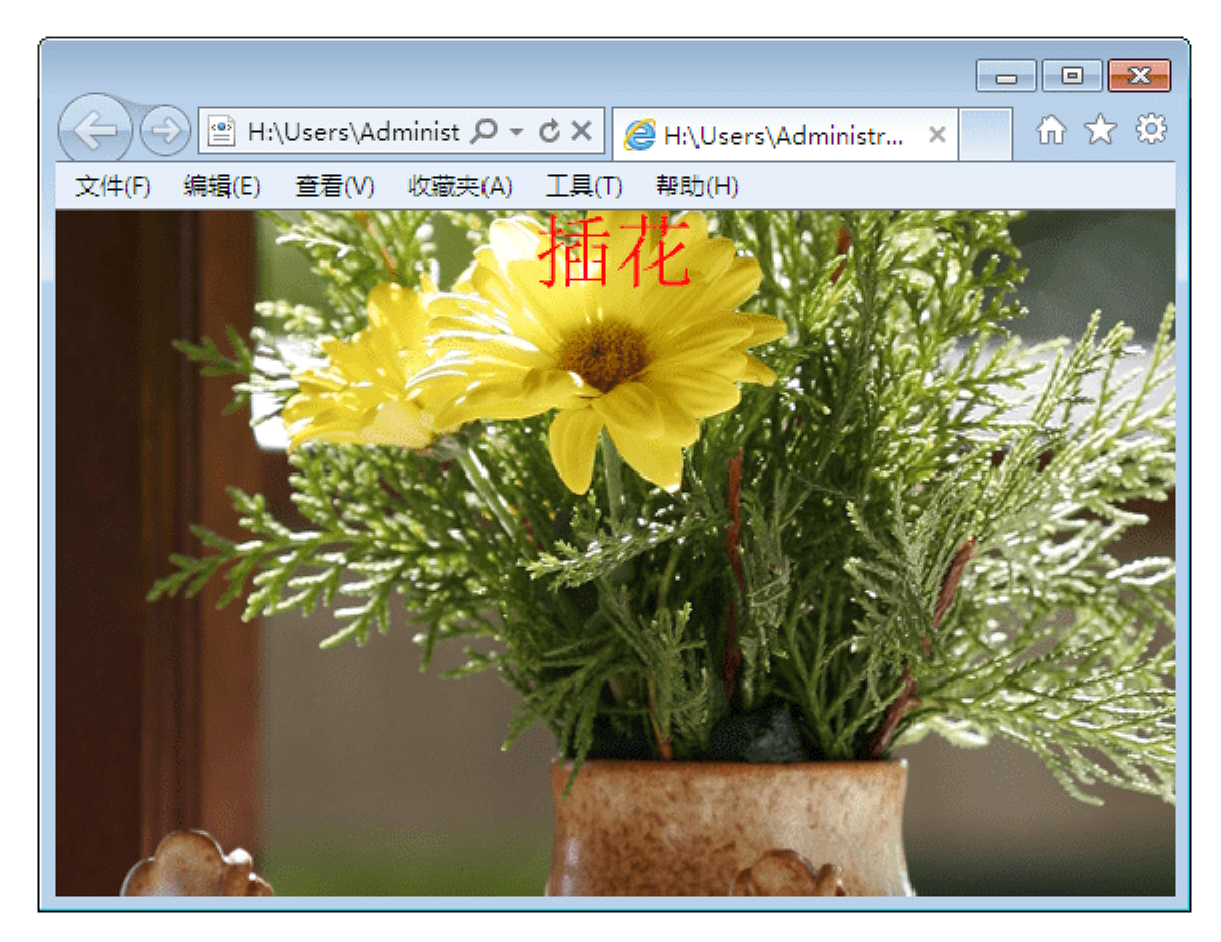


图 21-7　使用 CSS 定义 XML 背景

21.2.4　设置边框属性

在 CSS 中可以使用 border-style、border-width 和 border-color 这 3 个属性设定边框。页面元素的边框就是将元素内容及间隙包含在其中的边线，类似于表格的外边线。页面元素的边框可以 3 个方面来描述，即宽度、样式和颜色，这 3 个方面决定了边框所显示出来的外观。

【例 21.10】（实例文件：ch21\21.7.xml）

```
<?xml version="1.0" encoding="GB2312" ?>
<?xml-stylesheet href="20.7.css" type="text/css" ?>
<Border>
        <smallBorder>
      学校雷锋好榜样
      </smallBorder>
</Border>
```

【例 21.11】（实例文件：ch21\21.7.css）

```
Border{
    border-style:solid;
    border-width:15px;
    border-color:#123456;
    width:200px;
    height:150px;
    text-align:center;
}
smallBorder{
    font-size:20px;
    color:red;
}
```

在 Border 标记中设置边框显示样式，例如直线形显示，颜色为深蓝色，宽度为 15 像素，并且设置显示块的宽度为 200 像素，高度为 150 像素，边框内元素居中显示。在 smallBorder 标记中设置了字体大小和字体颜色。

在 IE 9.0 中浏览效果如图 21-8 所示，可以看到页面中显示了一个边框，边框中显示的是红色字体，其内容是“学习雷锋好榜样”。

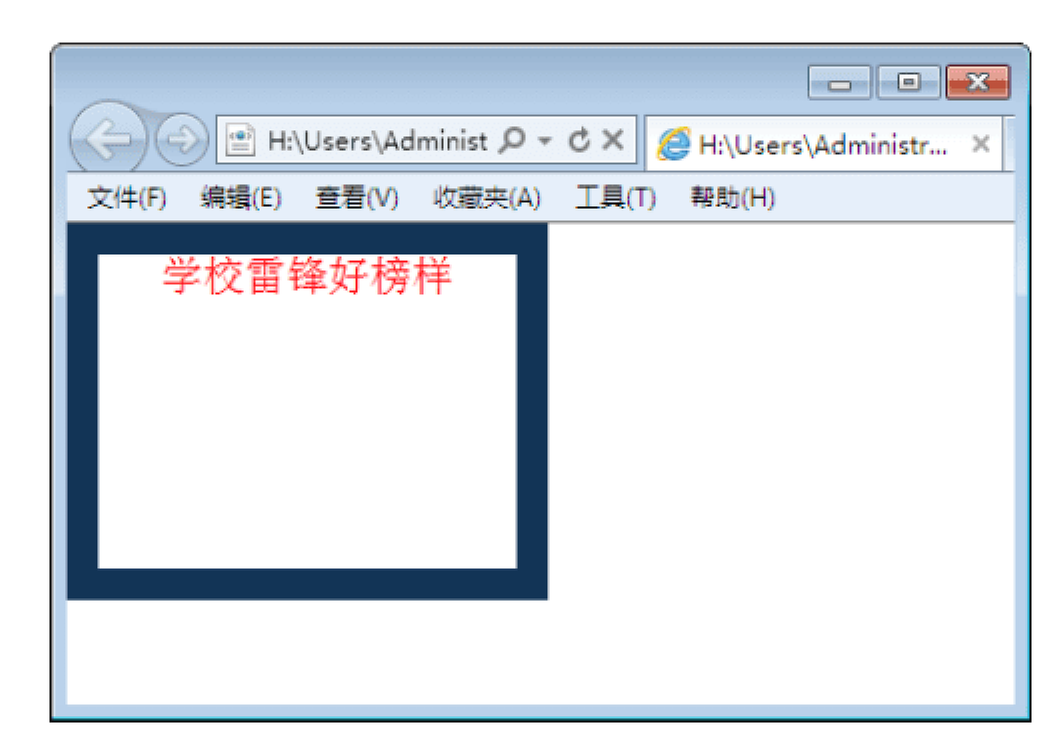


图 21-8　设置 XML 元素的边框

21.2.5　设置文本属性

在 CSS 中，提供了多种文本属性来实现对文本的控制，如 text-indent 属性、text-align 属性、white-space 属性、line-height 属性、vertical-align 属性、text-transform 属性和 text-decoration 属性。这些前面已经介绍过，这里不再赘述。利用上面这些属性，可以控制 XML 元素的显示样式。

【例 21.12】（实例文件：ch21\21.8.xml）

```
<?xml version="1.0" encoding="gb2312"?>
<?xml-stylesheet type="text/css" href="20.8.css"?>
<big>
  <one>健康</one>
<two>
<title>饮茶养生养颜 特殊时期慎饮茶</title>
<content>
金银花，味甘，性寒，具有清热解毒、疏散风热的作用。金银花为清热解毒之良药，既能清里热，又能散表热，临床上主要用于治疗各种痈肿疮毒、热毒血痢及温热病等。金银花药性偏寒，不适合长期饮用，仅适合在炎热的夏季暂时饮用以防治痢疾。
</content>
</two>
</big>
```

【例 21.13】（实例文件：ch21\21.8.css）

```
big{
  width:500px;
  border:#6600FF 1px solid;
  height:200px;
  font-size:12px;
  font-family:"幼圆";
  }
one{
```

```
        font-size:18px;
        width:500px;
        height:25px;
        line-height:25px;
        text-align:center;
        color:#FF3300;
        margin-top:5px;
        font-weight:800;
        text-decoration:underline;
        }
    title{
        margin:10px 0 10px 10px;
        display:block;
        color:#0033FF;
        font-size:14px;
        font-weight:800;
    text-align:center;
        }
    content{
        display:block;
        line-height:20px;
        width:490px;
        margin-left:10px;
        font-weight:800;
        text-indent:2em;
        }
```

上面的 CSS 代码分别定义了不同标记的显示样式，如宽度、高度、边框样式、字体大小、行高和是否带有下划线等。

在 IE 9.0 中浏览效果如图 21-9 所示，可以看到页面中显示了一个公告栏，栏中显示了不同颜色的字体，并且段落缩进两个单元格显示。

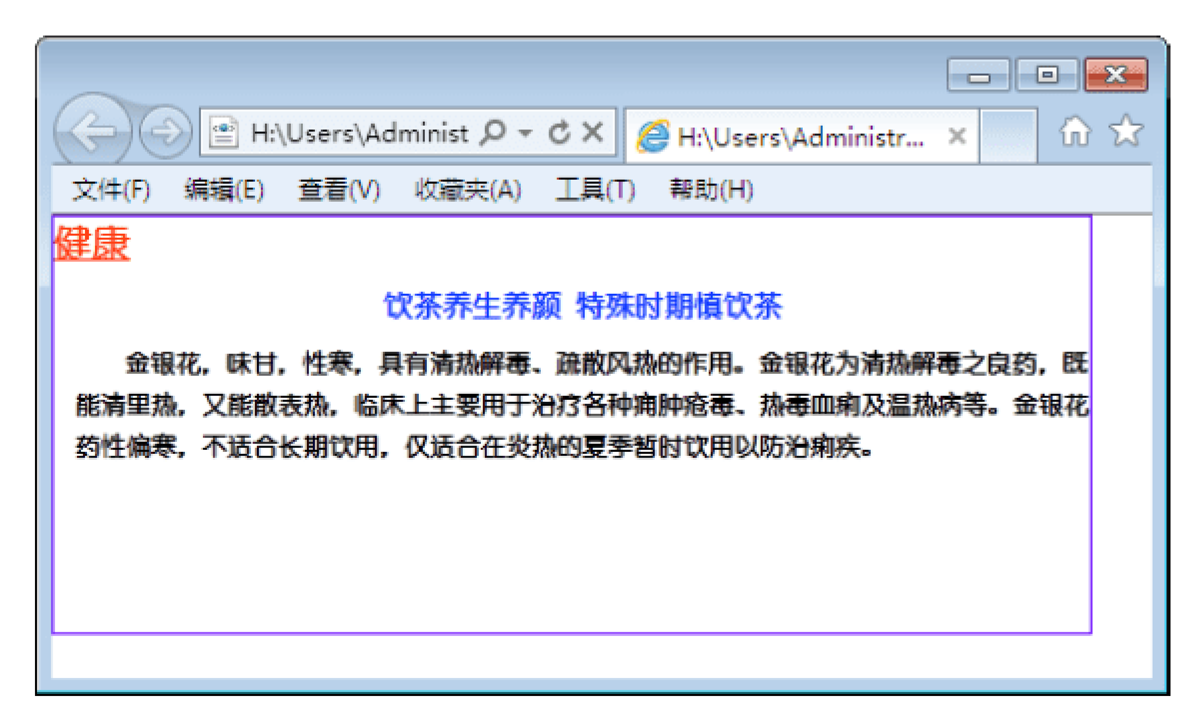


图 21-9 修饰 XML 文本

21.3　综合实战 1——招聘广告

CSS 结合 XML 文档，可以创建出多种多样的样式，如 HTML 页面中常见的招聘信息。本实例将结合前面学习的 XML 和 CSS 知识，创建一个基于 XML 文档的招聘广告。具体步骤如下所示。

01 分析需求。

创建一个招聘广告，就文件而言包括两个大部分，一个是 XML 文档，一个是 CSS 文件。XML 文档需要包含招聘标题、招聘内容和联系方式，而 CSS 文件需要针对 XML 文档标记进行样式修饰。实例完成后，效果如图 21-10 所示。

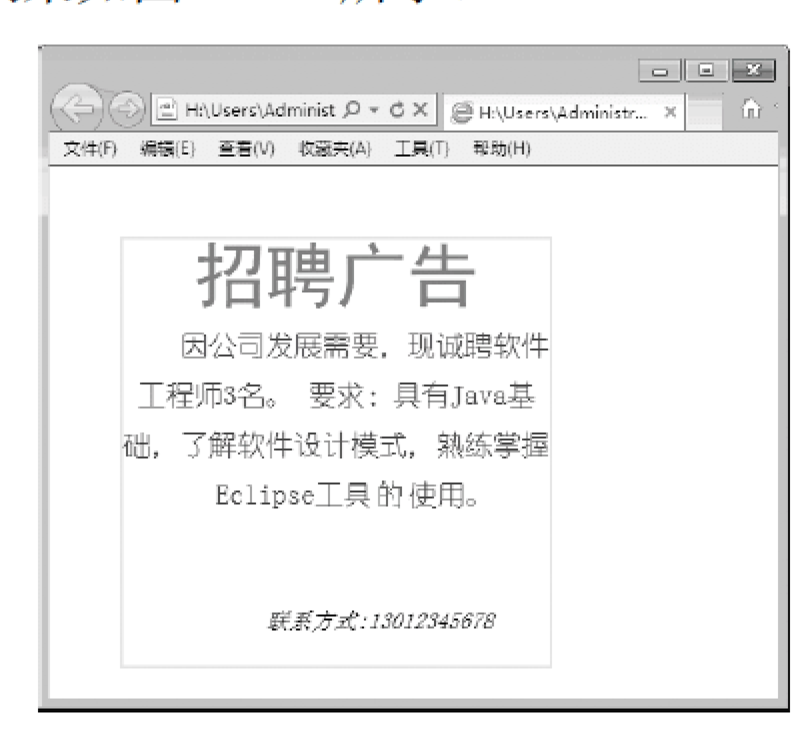


图 21-10　招聘广告显示

02 构建 XML 文档。

```
<?xml version="1.0" encoding="GB2312" ?>
<?xml-stylesheet href="20.9.css" type="text/css" ?>
<zhaopin>
    <title>
        招聘广告
    </title>
    <content>
      因公司发展需要，现诚聘软件工程师 3 名。
          要求：具有 Java 基础，了解软件设计模式，熟练掌握 Eclipse 工具的使用。
    </content>
    <address>
        联系方式:13012345678
    </address>
</zhaopin>
```

在 XML 文件中，有一个根标记<zhaopin>，下面包含 3 个子标记，分别为<title>标记、<content>标记和<address>标记。每个标记中分别存放相应的数据。

在 IE 9.0 中浏览效果如图 21-11 所示，可以看到页面中显示了一个树形结构，其元素中包

含不同的信息。

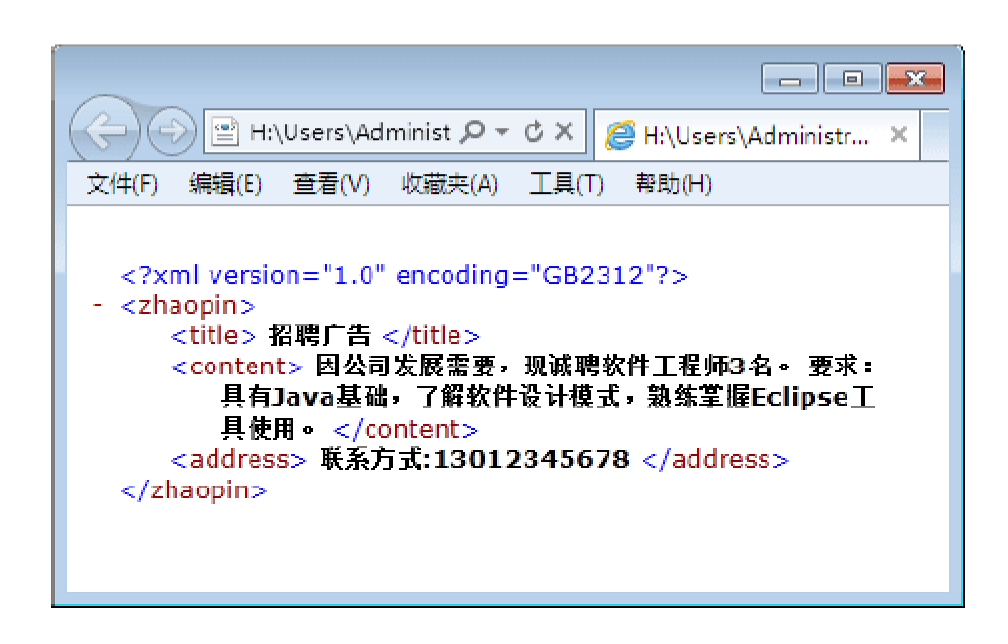


图 21-11　XML 文档显示

03 添加 CSS 样式文件，修饰整体样式。

```
zhaopin{
    display:block;
    position:absolute;
    top:50px;
    left:50px;
    width:300px;
    height:300px;
    border:2px #ddeecc solid;
    text-align:center;
}
```

上面的代码定义了 zhaopin 元素以块形式显示，使用绝对定位，坐标为（50,50），高度和宽度都是 300 像素。

在 IE 9.0 中浏览效果如图 21-12 所示，可以看到页面不再以树形结构显示，而是显示一个边框，边框内是 XML 文档的数据。

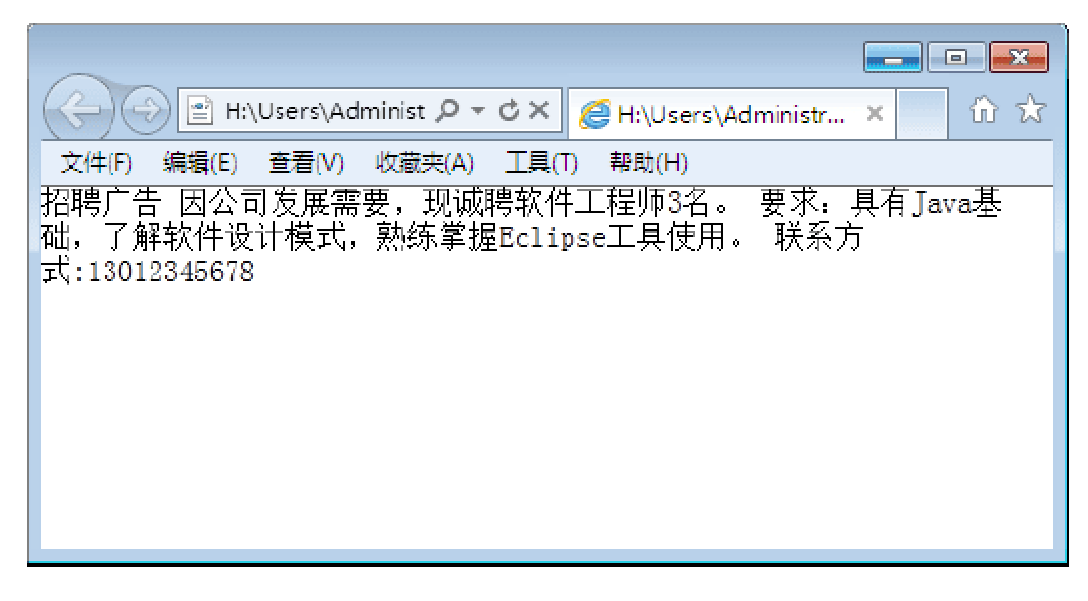


图 21-12　定义边框样式

04 添加 CSS 样式，修饰标题、内容和地址信息。

```
title{
     font-family:黑体;
     font-varanit:normal;
```

```
        font-size:50px;
      color:red;
    }
    content{
        display:block;
        font-family:"幼圆";
        font-varanit:normal;
        font-size:20px;
        color:blue;
        text-indent:2em;
        line-height:35px;
    }
    address{
      position:absolute;
      left:100px;
        top:260px;
      text-align:right;
      font-style:italic;
    }
```

在 CSS 文件中，针对上面不同的标记，设置相应的样式，如修饰字体、定位和块的大小。

在 IE 9.0 中浏览效果如图 21-13 所示，可以看到页面边框内，标题以红色字体显示，正文以蓝色字体显示，联系方式以斜体显示。

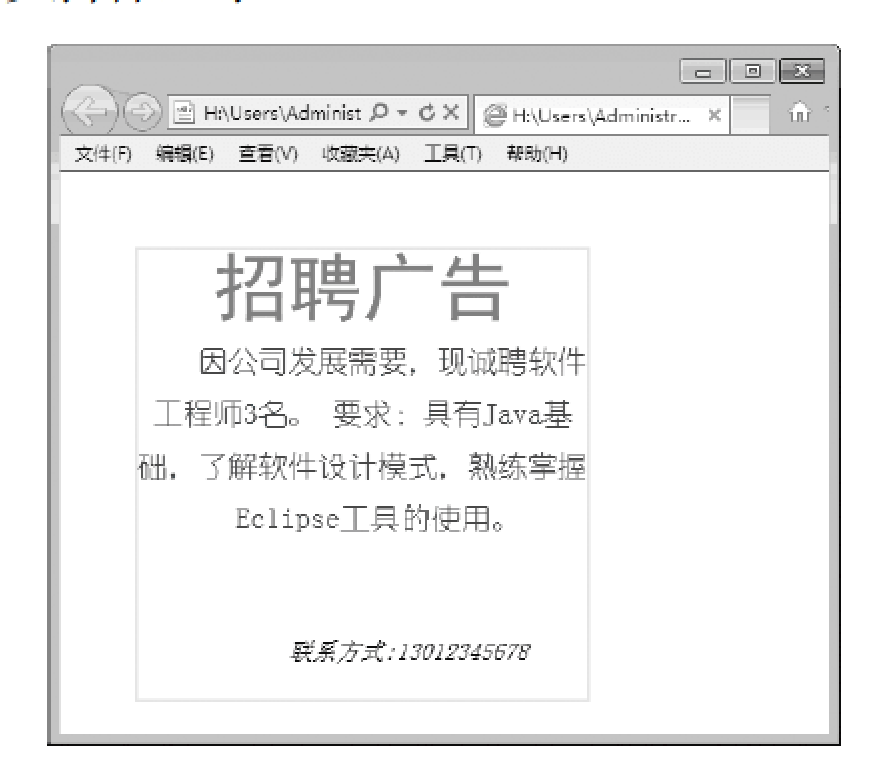


图 21-13　最终效果

21.4　综合实战 2——书目清单

在网页中，使用表格显示数据是最常见的一种方式，本实例将使用 XML 和 CSS 模拟表格效果，创建一个书目清单。具体步骤如下所示。

01 分析需求。

创建一个具有表格效果的页面，是利用元素的边框效果完成的。针对 XML 标记都设定一个边框效果，相近的边框重叠就可以了。实例完成后，效果如图 21-14 所示。

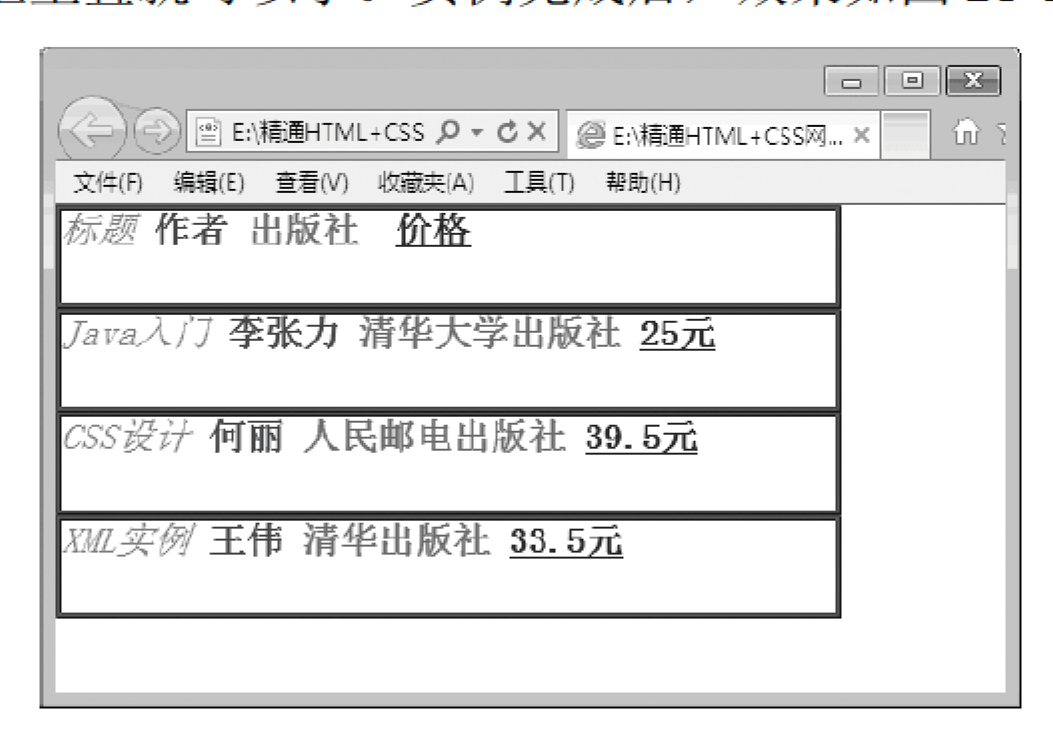


图 21-14 模拟表格效果

02 创建 XML 文档。

```
<?xml version="1.0" encoding="GB2312"?>
<?xml-stylesheet type="text/css" href="20.10.css"?>
<INVENTORY>
        <image></image>
        <BOOK>
            <TITLE>标题</TITLE>
            <AUTHOR>作者</AUTHOR>
            <BINDING>出版社</BINDING>
            <PRICE>价格</PRICE>
        </BOOK> <BOOK>
            <TITLE>Java 入门</TITLE>
            <AUTHOR>李张力</AUTHOR>
            <BINDING>清华大学出版社</BINDING>
            <PRICE>25 元</PRICE>
        </BOOK>
        <BOOK>
            <TITLE>CSS 设计</TITLE>
            <AUTHOR>何丽</AUTHOR>
            <BINDING>人民邮电出版社</BINDING>
            <PRICE>39.5 元</PRICE>
        </BOOK>
        <BOOK>
             <TITLE>XML 实例</TITLE>
             <AUTHOR>王伟</AUTHOR>
             <BINDING>清华出版社</BINDING>
            <PRICE>33.5 元</PRICE>
        </BOOK>
```

```
</INVENTORY>
```

XML 文件中的第二行，使用 xml-stylesheet 标记将 CSS 文件引入到 XML 文件中，下面创建了 BOOK 标记及其子标记 TITLE、AUTHOR、PRICE 等。每个标记都包含不同的数据。

在 IE 9.0 中浏览效果如图 21-15 所示，可以看到页面以树形结构显示 XML 文件。

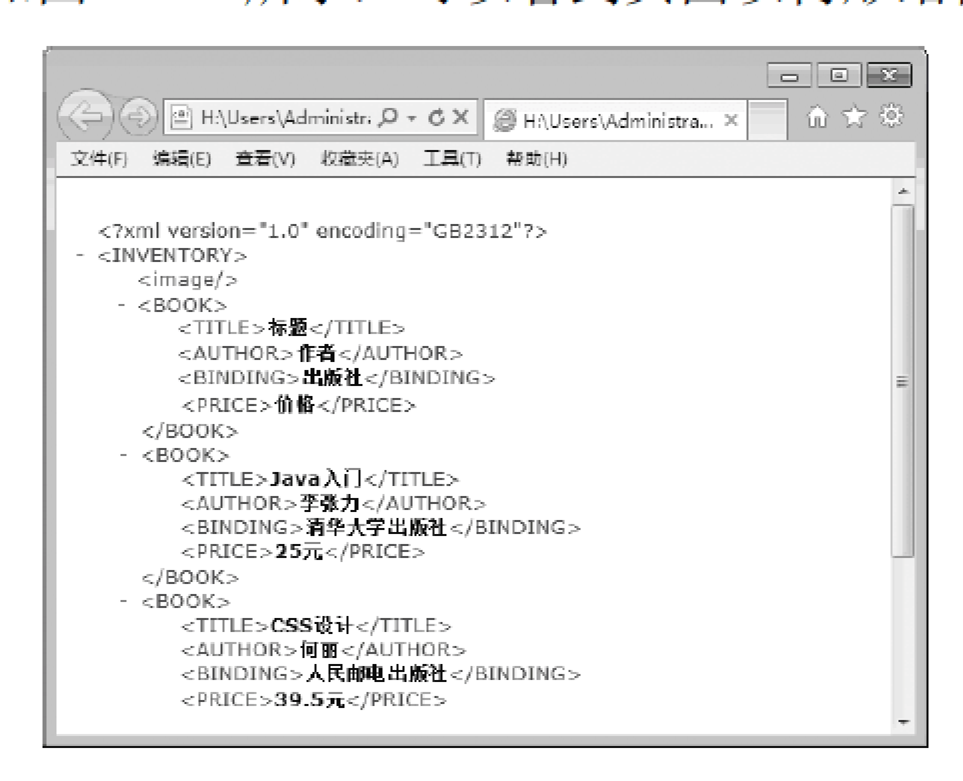


图 21-15　图书目录显示

03 创建 CSS 文件，修饰指定元素。

```
BOOK{
        display:block;
        font-size:20px;
        text-align:left;
        border-style:ridge;
        color:black;
        width:420px;
         height:50px;
}
TITLE{
        font-style:italic;
        color:#009900;
}
AUTHOR{
        font-weight:bold;
        color:BLUE;
}
BINDING{
        font-weight:bold;
        color:GREEN;
}
PRICE{
       font-weight:bold;
       color:#000099;
```

```
        text-align: left;
        font-weight: bold;
        text-decoration:underline;
    }
```

从代码中，可以看出针对 XML 文件中的标记，定义了不同的样式。其思路是先定义 BOOK 标记样式，然后再定义其子标记样式。

在 IE 9.0 中浏览效果如图 21-16 所示，可以看到页面中以表格方式，显示不同的数据，其数据以不同的颜色分开。

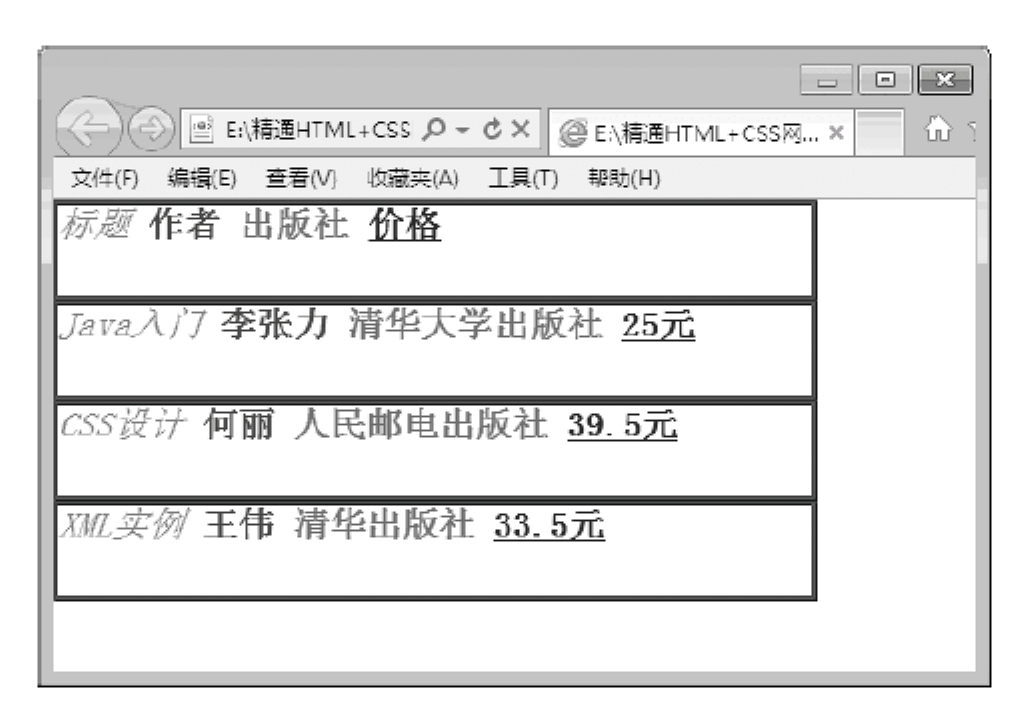


图 21-16　最终效果

21.5　综合实战 3——图文混排页面

图文搭配布局是显示的永恒话题，通过文字进行介绍，通过图形进行说明，两者相得益彰，互为补充。本实例使用 XML 文档结合 CSS 文件完成图文混搭的布局。具体步骤如下所示。

01 分析需求。

将图形和文字放在一起进行布局，需要将图形放在一个指定的位置显示，最好定义其显示大小，文字可以环绕图片，也可以放在图片一侧显示。实例完成后，效果如图 21-17 所示。

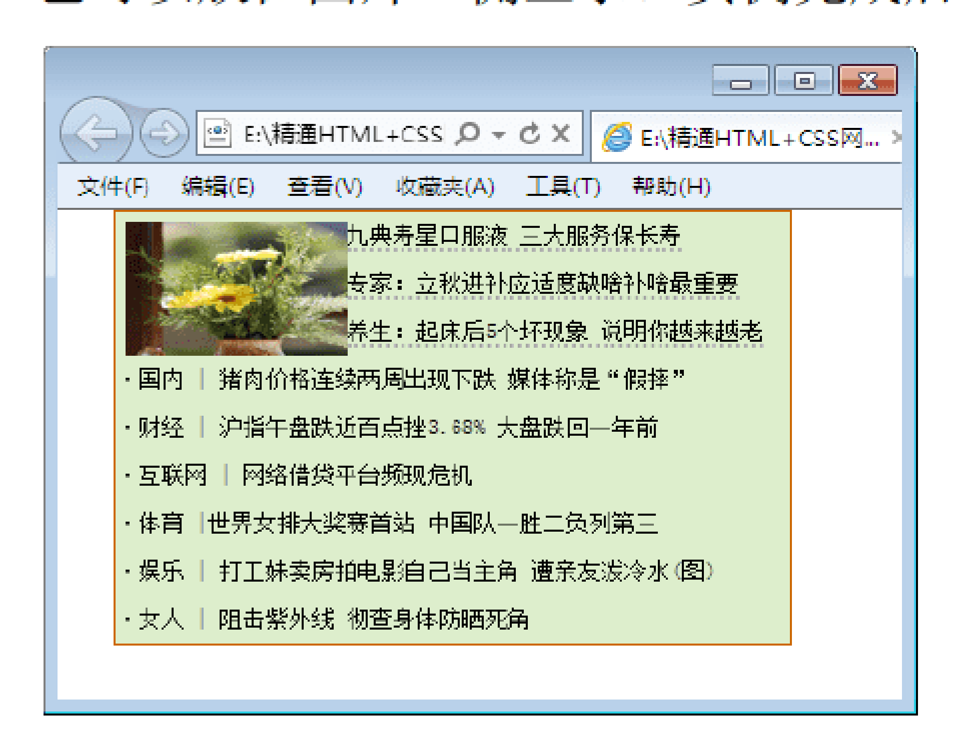


图 21-17　图文混排

02 构建 XML 文档。

```
<?xml version="1.0" encoding="gb2312"?>
<?xml-stylesheet type="text/css" href="20.11.css"?>
<xinwen>
  <content>
    <right>
      <img></img>
      九典寿星口服液  三大服务保长寿
      <br />
      专家：立秋进补应适度缺啥补啥最重要
      <br />
      养生：起床后 5 个坏现象  说明你越来越老
      <br/>
    </right>
    • 国内 | 猪肉价格连续两周出现下跌  媒体称是“假摔”
    <br />
    • 财经 | 沪指午盘跌近百点挫 3.68%  大盘跌回一年前
    <br />
    • 互联网 | 网络借贷平台频现危机
    <br />
    • 体育  |世界女排大奖赛首站  中国队一胜二负列第三
    <br />
    • 娱乐 | 打工妹卖房拍电影自己当主角  遭亲友泼冷水(图)
    <br />
    • 女人 | 阻击紫外线  彻查身体防晒死角
  </content>
</xinwen>
```

在 XML 文件中，首先定义了根标记 xinwen，然后定义了 content 标记，content 标记中包含 right、img 和 br 标记。

在 IE 9.0 中浏览效果如图 21-18 所示，可以看到页面中显示了一个树形结构，其元素中包含了不同的文本数据。

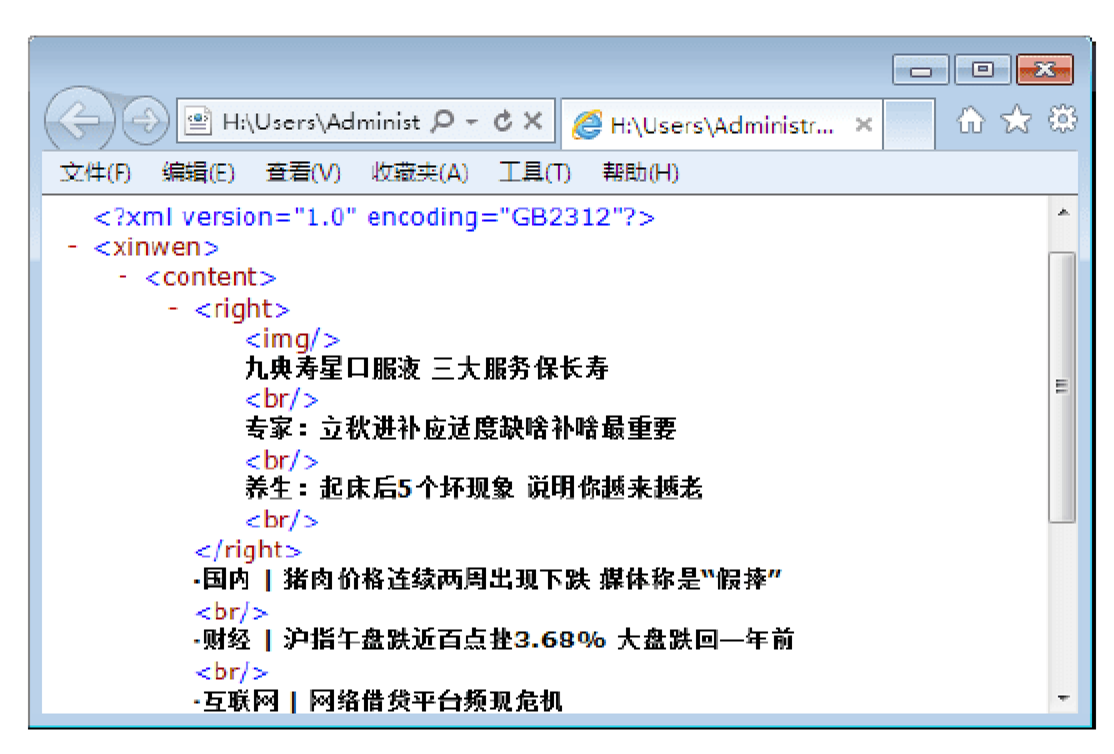


图 21-18　图文混排树形结构

03 添加 CSS 样式，修饰 xinwen、br 和 content 元素。

```
xinwen{
    width:400px;
    font-size:12px;
    font-family:"宋体";
    margin:0 auto 0 auto;
    }
br{
    display:block;
    }
content{
    width:350px;
    border:#CC6600 1px solid;
    float:left;
    line-height:25px;
    margin-left:10px;
    background-color:#ddeecc;
    }
```

在 CSS 文件中，创建了不同的标记选择器，针对 XML 文件标记进行修饰，即设置字体、边框、位置和背景属性。需要注意的是，由于使用了 XML 文档，所有
标记不能像在 HTML 里那样实现换行的效果，这里需要为
定义 CSS 样式，才能实现正文里的换行效果。

在 IE 9.0 中浏览效果如图 21-19 所示，可以看到页面中显示了黄色边框，边框内显示了相关列表信息。

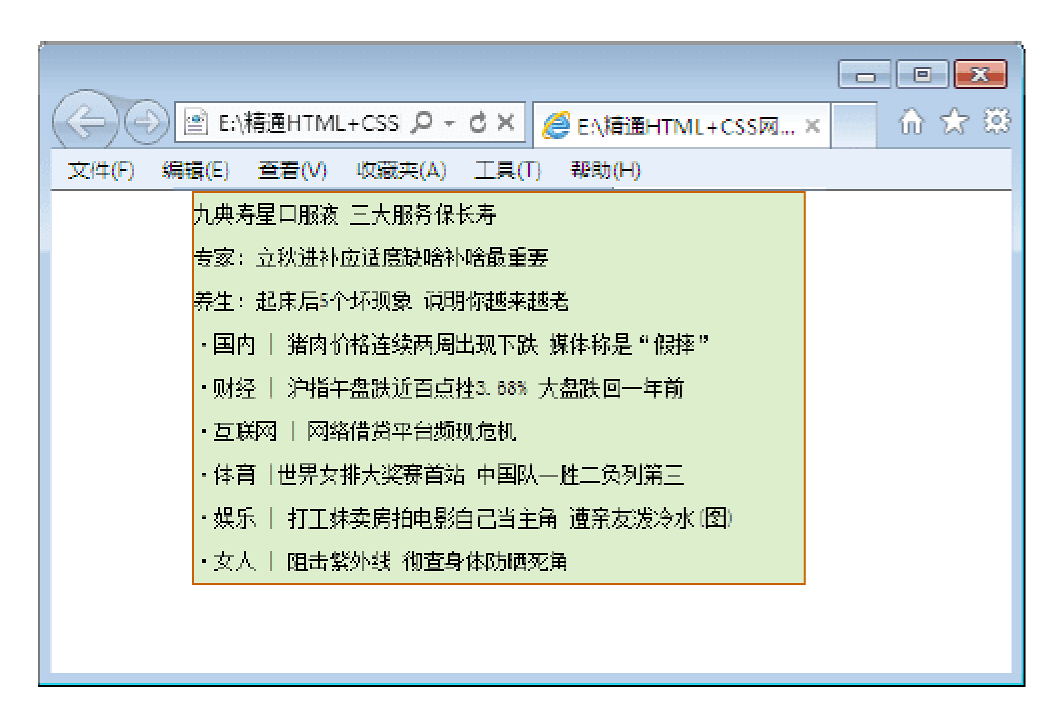


图 21-19　修饰指定标记

04 添加 CSS 样式，修饰 img 和 right 元素。

```
img{
    width:115px;
    height:70px;
    background-image:url(003.jpg);
    float:left;
```

```
    margin:5px 0 0 5px;
    }
right{
    border-bottom:#999999 dotted 2px;
    }
```

上面的 CSS 代码中定义了 img 元素显示样式，如宽度、高度、背景图片、左浮动和外边距距离等。

在 IE 9.0 中浏览效果如图 21-20 所示，可以看到页面中边框内显示了一张图片。

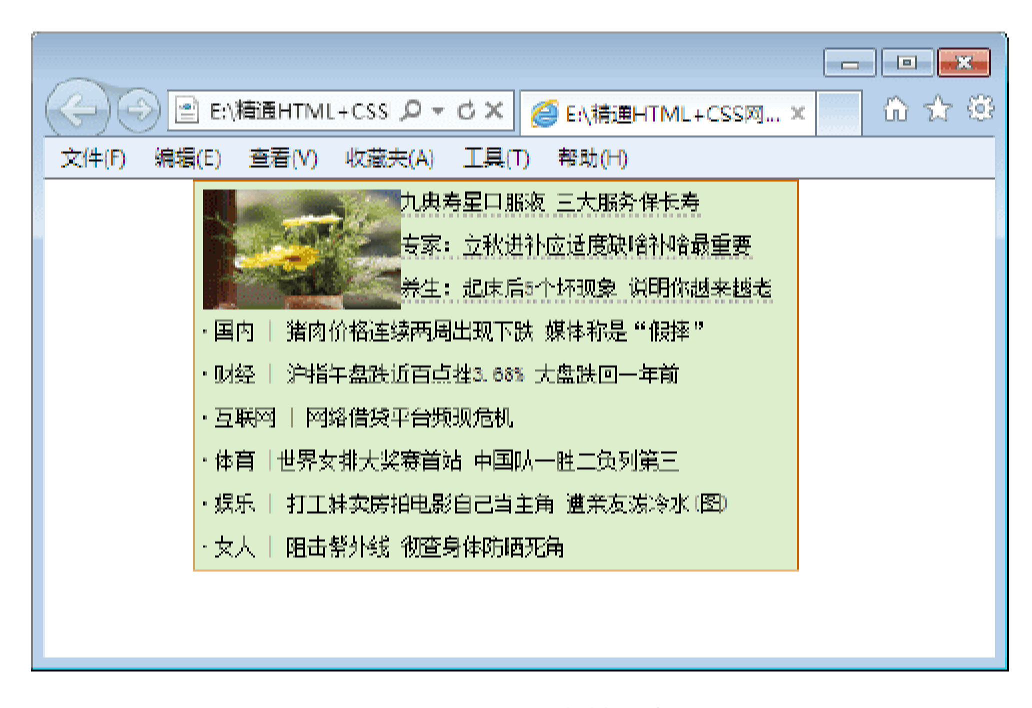


图 21-20　最终效果

21.6　综合实战 4——古诗欣赏

在一些古典风格的网站中，通常有很多漂亮的古诗页面。本实例将模拟这种效果。具体步骤如下所示。

01 分析需求。

如果要展示古诗内容，需要使用绝对定位的方法，为 XML 文档的段落文字排版。同样也可以导入一个漂亮的背景，配合文字展示。实例完成后，效果如图 21-21 所示。



图 21-21　古诗欣赏页面

02 创建 XML 文档。

```
<?xml version="1.0" encoding="gb2312"?>
<?xml-stylesheet type="text/css" href="20.12.css"?>
<qiu>   <title>立秋</title>
  <author>宋  刘翰 </author>
  <content>
  乳鸦啼散玉屏空，一枕新凉一扇风。<br/>
  睡起秋色无觅处，满阶梧桐月明中 。<br/>
  </content></qiu>
```

XML 文档的内容非常简单，创建了一个根标记 qiu，及其子标记 title、author 和 content。xml-stylesheet 表示引入一个 CSS 文件。在 IE 9.0 中浏览效果如图 21-22 所示，可以看到页面中显示了一个树形结构。

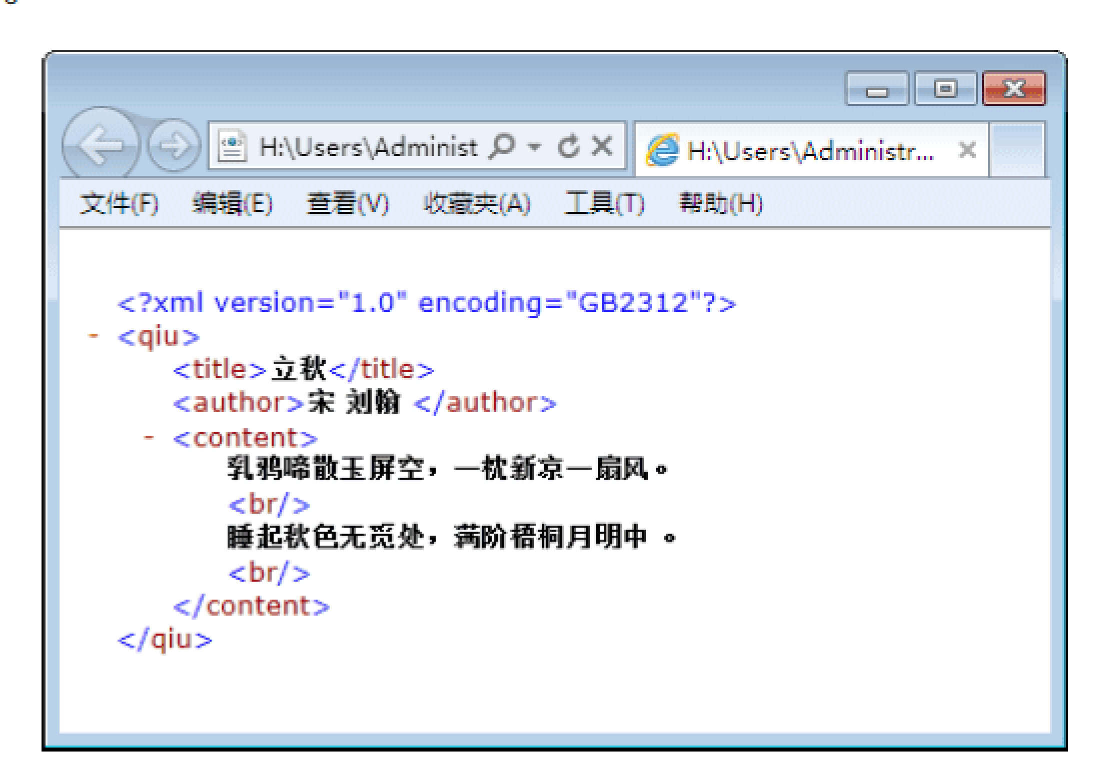


图 21-22　古诗树形结构

03 创建 CSS 文件，修饰 qiu 元素。

```
qiu{
  margin:0px;
  background:url(lan.jpg) no-repeat;   /*添加背景图片 */
  width:460px;
  height:320px;
  position:absolute;   /* 绝对定位 */}
```

上面的代码定义了 qiu 标记显示样式，如外边距、背景图片、宽度、高度和绝对定位。在 IE 9.0 中浏览效果如图 21-23 所示，可以看到页面中显示了一段文字，无任何格式。

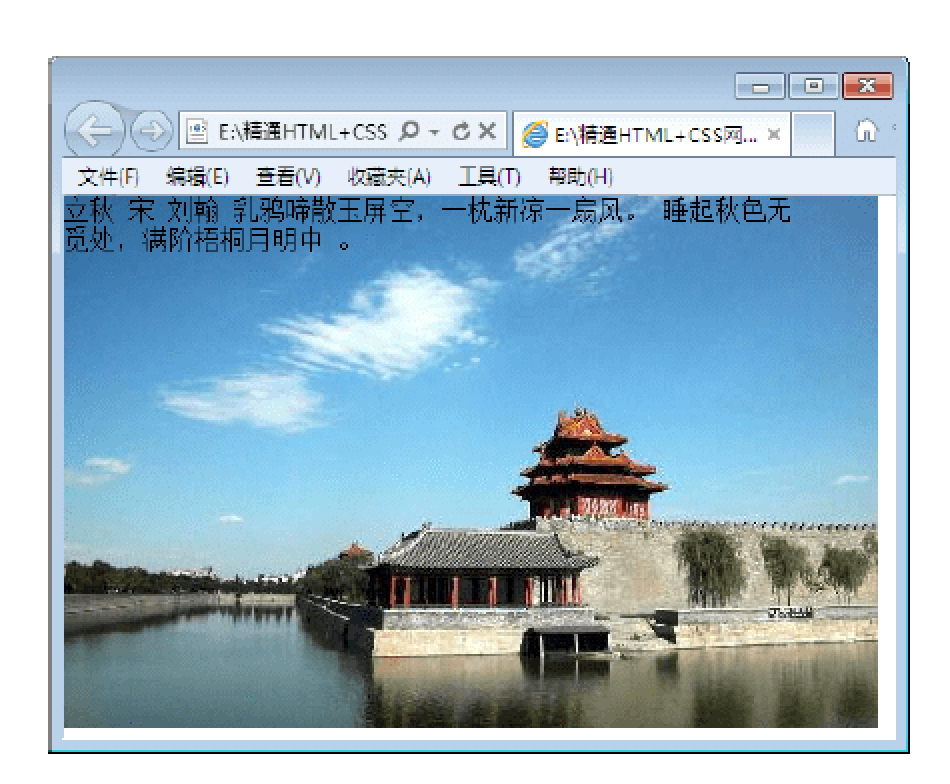


图 21-23　定义背景

04 添加 CSS 代码，修饰 title 和 author 元素。

```
title{
    font-size:30px;
    color:green;
    position:absolute;
    left:140px;
    top:20px;
}author{
    font-size:18px;
    color:#4f2b00;
    position:absolute;
    left:160px;
    top:50px;
}
```

上面的 CSS 代码中，定义了 title 标记和 author 标记的显示样式，如字体大小、颜色、绝对定位和坐标位置。

在 IE 9.0 中浏览效果如图 21-24 所示，可以看到页面中的标题和作者信息，在指定位置显示。

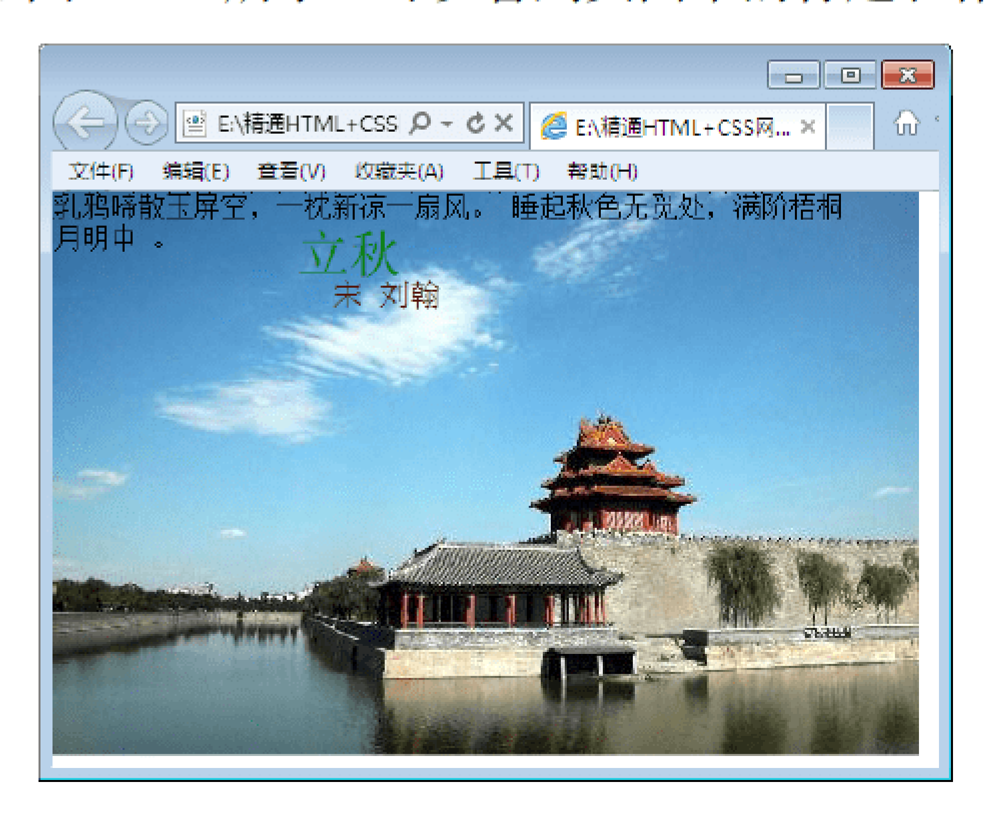


图 21-24　定义标题和作者信息

05 添加 CSS 代码，修饰 cotent 元素。

```
content{
  position:absolute;  /* 绝对定位 */
  font-size:24px;
  line-height:30px;  /* 行间距 */
  left:40px;
  top:70px;
              font-family:"幼圆";}
br{   display:block;     /* 让诗句分行显示 */}
```

在 CSS 文件中，定义了 content 元素的显示样式，如字体大小、行高、坐标位置等。这里的
标记也起到了换行作用，当将标记的 display 属性定义为 block 时，可以得到换行的效果。在 IE 9.0 中浏览效果如图 21-25 所示，可以看到页面中的古诗在指定位置显示，并且标题颜色为绿色，作者颜色为黄色。



图 21-25 最终效果

21.7 专家解惑

1. 背景色和背景图片放在一起定义好吗？

在设定背景图片时，最好同时也设定背景色，这样当背景图片因某种原因无法正常显示时，可以使用背景色来代替。当然，如果正常显示，背景图片会覆盖背景色的。

2. XML 和 HTML 文件有什么相同之处和不同之处？

HTML 和 XML 都是从 SGML 发展而来的标记语言，因此，它们有些共同点，如相似的语法和标记。不过 HTML 是 SGML 定义下的一个描述性语言，只是 SGML 的应用。而 XML 是 SGML 的简化版本，是 SGML 的一个子集。

XML 是用来存放数据的，XML 不是 HTML 的替代品，XML 和 HTML 是两种用途不同的语言。XML 是被设计用来描述数据的。HTML 只是一个显示数据的标记语言。

第 22 章

CSS与Ajax的综合应用

Ajax 是目前新兴的一项网络技术。确切地说，Ajax 是一种用于创建更好、更快以及交互性更强的 Web 应用程序的技术。它能使浏览器为用户提供更为自然的浏览体验，就像在使用桌面应用程序一样。

22.1 Ajax 概述

Ajax 是一项很有生命力的技术，它的出现引发了 Web 应用的新革命。目前，网络上的许多站点，使用 Ajax 技术的还很少，但是，可以预见在不远的将来，Ajax 技术会成为整个网络的主流。

22.1.1 什么是 Ajax

Ajax 的全称为 Asynchronous JavaScript And XML，是一种 Web 应用程序客户机技术，它结合了 JavaScript、层叠样式表（Cascading Style Sheets，CSS）、HTML、XMLHttpRequest 对象和文档对象模型（Document Object Model，DOM）多种技术。运行在浏览器上的 Ajax 应用程序，以一种异步的方式与 Web 服务器通信，并且只更新页面的一部分。通过利用 Ajax 技术，可以提供基于浏览器的丰富的用户体验。

Ajax 让开发者在浏览器端更新被显示的 HTML 内容而不必刷新页面。换句话说，Ajax 可以使基于浏览器的应用程序更具交互性而且更类似于传统型桌面应用程序。Google 的 Gmail 和 Outlook Express 就是两个使用 Ajax 技术的例子。而且，Ajax 可以用于任何客户端脚本语言中，包括 JavaScript、JScript 和 VBScript。

下面给出一个简单的实例，来具体了解一下什么是 Ajax。

【例 22.1】（实例文件：ch22\22.1.html）

本实例从一个简单的角度入手，实现客户端与服务器的异步通信，获取“你好，Ajax”的数据，并在不刷新页面的情况下将获得的“你好，Ajax”数据显示到页面上。

具体步骤如下。

01 使用记事本创建 HelloAjax.jsp 文件。代码如下:

```
<%@ page language="java" pageEncoding="gb2312"%>
```

```
<html>
  <head>
    <title>第一个 Ajax 实例</title>
    <style type="text/css">
      <!--
       body {
             background-image: url(images/img.jpg);
       }
      -->
    </style>
  </head>
  <script type="text/javascript">
   ………………//省略了 script 代码
  </script>
  <body><br>
    <center>
       <button onclick="hello()">Ajax</button>
       <P id="p">
           单击按钮后你会有惊奇的发现哟！
       </P>
    </center>
  </body>
</html>
```

JavaScript 代码嵌入在标签<script></script>之内，这里定义了一个函数 hello()，这个函数是通过一个按钮来驱动的。

02 在步骤 1 省略的代码部分创建 XMLHttpRequest 对象，创建完成后把此对象赋值给 XMLHttp 变量。为了获得多种浏览器的支持，应使用 createXMLHttpRequest()函数为多种浏览器创建 XMLHttpRequest 对象，代码如下：

```
var xmlHttp=false;
function createXMLHttpRequest()
{
     if (window.ActiveXObject)                    //在 IE 浏览器中创建 XMLHttpRequest 对象
     {
          try{
           xmlHttp=new ActiveXObject("Msxml2.XMLHTTP");
        }
          catch(e){
               try{
                 xmlHttp = new ActiveXObject("Microsoft.XMLHTTP");
               }
```

```
                    catch(ee){
                    xmlHttp=false;
                  }
            }
      }
      else if (window.XMLHttpRequest)                    //在非 IE 浏览器中创建 XMLHttpRequest 对象
      {
           try{
              xmlHttp = new XMLHttpRequest();
           }
           catch(e){
              xmlHttp=false;
           }
      }
}
```

03 在步骤 1 省略的代码部分再定义 hello()函数，hello()函数为要与之通信的服务器资源创建一个 URL，“xmlHttp.onreadystatechange=callback;”定义了 JavaScript 回调函数，一旦响应它就自动执行，而“xmlHttp.open("post","HelloAjaxDo.jsp",true);”中所指定的“true”标志说明想要异步执行该请求，如果没有指定则默认为“true”。代码如下:

```
function hello()
{
     createXMLHttpRequest();     //调用创建 XMLHttpRequest 对象的方法
     xmlHttp.onreadystatechange=callback;     //设置回调函数
     xmlHttp.open("post","HelloAjaxDo.jsp",true);          //向服务器端 HelloAjaxDo.jsp 发送请求
     xmlHttp.setRequestHeader("Content-Type","application/x-www-form-urlencoded;charset=gb2312");
     xmlHttp.send(null);
     function callback()
     {
          if(xmlHttp.readyState==4)
          {
               if(xmlHttp.status==200)
               {
                var data= xmlHttp.responseText;
                var pNode=document.getElementById("p");
                pNode.innerHTML=data;
               }
          }
     }
}
```



函数 callback()是回调函数，它首先检查 XMLHttpRequest 对象的整体状态以保证它已经完成（readyStatus==4），然后根据服务器的设定询问请求状态。如果一切正常，即（status==200），就使用“var data=xmlHttp.responseText;”取得返回的数据，用 innerHTML 属性重写 DOM 的“pNode”节点的内容。

JavaScript 的变量类型使用的是弱类型，都使用 var 来声明。document 对象就是文档对应的 DOM 树。通过“document.getElementById("p");”可以用标签的 id 值来取得此标签的一个引用（树的节点）；“pNode.innerHTML=str;”为节点添加内容，这样就覆盖了节点的原有内容，如果不想覆盖可以使用“pNode.innerHTML+=str;”来追加内容。

04 通过步骤 3 可以知道要异步请求的是 HelloAjaxDo.jsp，下面需要创建此文件，代码如下：

```
<%@ page language="java" pageEncoding="gb2312"%>
<%
  out.println("你好，Ajax");
%>
```

05 将上述文件保存在 Ajax 站点下，启动 Tomcat 服务器打开浏览器，在地址栏输入“http://localhost:8080/Ajax/HelloAjax.jsp”，单击【转到】按钮，结果如图 22-1 所示。



图 22-1 会变的页面

06 单击 Ajax 按钮，显示页面如图 22-2 所示，注意按钮下内容的变化。



图 22-2 动态改变页面

22.1.2 Ajax 的关键元素

Ajax 不是单一的技术，而是 4 种技术的集合，要灵活地运用 Ajax，必须深入了解这些不同的技术，表 22-1 中列出了这些技术，以及它们在 Ajax 中所扮演的角色。

表 22-1 Ajax 涉及的技术

名称	描述
JavaScript	JavaScript 是通用的脚本语言，用来嵌入在某种应用之中。Web 浏览器中嵌入的 JavaScript 解释器允许通过程序与浏览器的很多内建功能进行交互。Ajax 应用程序是使用 JavaScript 编写的
CSS	CSS 为 Web 页面元素提供了一种可重用的可视化样式的定义方法。它提供了简单而又强大的方法，以一致的方式定义和使用可视化样式。在 Ajax 应用中，用户界面的样式可以通过 CSS 独立修改
DOM	DOM 以一组可以使用 JavaScript 操作的可编程对象展现出 Web 页面的结构。通过使用脚本修改 DOM，Ajax 应用程序可以在运行时改变用户界面，或者高效地重绘页面中的某个部分
XMLHttpRequest 对象	XMLHttpRequest 对象允许 Web 程序员从 Web 服务器以后台活动的方式获取数据。数据格式通常是 XML，但是也可以很好地支持任何基于文本的数据格式

Ajax 的 4 种技术当中，CSS、DOM 和 JavaScript 都是很早就出现的技术，它们以前结合在一起称为动态 HTML，即 DHTML。

Ajax 的核心是 JavaScript 对象 XmlHttpRequest。该对象在 Internet Explorer 5 中首次引入，它是一种支持异步请求的技术。简而言之，XmlHttpRequest 使您可以使用 JavaScript 向服务器提出请求并处理响应，而不阻塞用户。

22.1.3 CSS 在 Ajax 应用中的地位

CSS 在 Ajax 中主要用于美化网页，是 Ajax 的美术师。无论 Ajax 的核心技术采用什么形式，任何时候显示在用户面前的都是一个页面，就需要 CSS 对显示在用户浏览器上的界面进行美化。

如果用户在浏览器中查看页面的源代码，就可以看到众多的<div>块以及 CSS 属性占据了源代码的大部分，如图 22-3 所示。从这一点可见 CSS 在页面美化方面的重要性。

http://www.114la.com/?114la - 原始源

文件(F)　编辑(E)　格式(O)

```
<title>114啦网址导航[Www.114La.Com]_最实用的上网主页_上网就上114啦! </title>
<script type="text/javascript">var BaiduTn = {tn:"ylmf_4_pg",ch:"4" };</script>
<script type="text/javascript" src="public/home_new/js/base.js" charset="gb2312"></script>

<link id="js_oStyle" href="public/home_new/css/index.css?v0308" rel="stylesheet"
type="text/css" />

<link rel="stylesheet" id="js_skinStyle" href="public/home_new/css/skin/blue.css" />

<script type="text/javascript">

 /**处理cookie(包括宽屏,旧版,皮肤)**/

(function(){

 if (Cookie.get('layout')) {

  window.location.href = 'kp.html';

 }
```

图 22-3　源文件中的 CSS 代码

22.2　Ajax 快速入门

Ajax 作为一项新技术，结合 4 种不同的技术，实现了客户端与服务器端的异步通信，并且对页面实现局部更新，大大提高了浏览器的速度。

22.2.1　全面剖析 XMLHttpRequest 对象

XMLHttpRequest 对象是当今所有 Ajax 和 Web 2.0 应用程序的技术基础。尽管软件经销商和开源社团现在都在提供各种 Ajax 框架，以进一步简化 XMLHttpRequest 对象的使用；但是，我们仍然很有必要理解这个对象的详细工作机制。

1．XMLHttpRequest 概述

Ajax 利用一个构建到所有现代浏览器内部的对象 XMLHttpRequest 来实现发送和接收 HTTP 请求与响应信息。一个经由 XMLHttpRequest 对象发送的 HTTP 请求并不要求页面中拥有或回寄一个＜form＞元素。

在微软 IE 5 中，作为一个 ActiveX 对象形式引入了 XMLHttpRequest 对象。其他认识到这一对象重要性的浏览器制造商也都纷纷在其浏览器内实现了 XMLHttpRequest 对象，但是作为一个本地 JavaScript 对象而不是作为一个 ActiveX 对象实现的。

如今，在认识到实现这一类型的价值及安全性特征之后，微软已经在其 IE 7 中把 XMLHttpRequest 实现为一个窗口对象属性。幸运的是，尽管其实现细节不同，但是，所有的浏览器实现都具有类似的功能，并且实质上是相同方法。目前，W3C 组织正在努力进行 XMLHttpRequest 对象的标准化。

2．XMLHttpRequest 对象的属性和事件

XMLHttpRequest 对象包括各种属性、方法和事件，以便于脚本处理并控制 HTTP 请求与

响应。下面对此进行详细的讨论。

- readyState 属性

当 XMLHttpRequest 对象把一个 HTTP 请求发送到服务器时将经历若干种状态，一直等待直到请求被处理，它才接收一个响应。这样以来，脚本才正确响应各种状态，XMLHttpRequest 对象暴露描述对象的当前状态的是 readyState 属性，如表 22-2 所示。

表 22-2　XMLHttpRequest 对象的 readyState 属性值列表

readyState 取值	描述
0	描述一种“未初始化”状态；此时，已经创建一个 XMLHttpRequest 对象，但是还没有初始化
1	open()方法并且 XMLHttpRequest 已经准备好把一个请求发送到服务器
2	描述一种“发送”状态；此时，已经通过 send()方法把一个请求发送到服务器端，但是还没有收到一个响应
3	描述一种“正在接收”状态；此时，已经接收到 HTTP 响应头部信息，但是消息体部分还没有完全接收结束
4	描述一种“已加载”状态；此时，响应已经被完全接收

- onreadystatechange 事件

无论 readyState 值何时发生改变，XMLHttpRequest 对象都会激发一个 onreadystatechange 事件。其中，onreadystatechange 属性接收一个 EventListener 值，该值向该方法指示无论 readyState 值何时发生改变，该对象都将激活。

- responseText 属性

这个 responseText 属性包含客户端接收到的 HTTP 响应的文本内容。当 readyState 值为 0、1 或 2 时，responseText 包含一个空字符串。当 readyState 值为 3（正在接收）时，响应中包含客户端还未完成的响应信息。当 readyState 为 4（已加载）时，该 responseText 包含完整的响应信息。

- responseXML 属性

responseXML 属性用于当接收到完整的 HTTP 响应时描述 XML 响应；此时，Content-Type 头部指定 MIME（媒体）类型为 text/xml、application/xml 或以+xml 结尾。如果 Content-Type 头部并不包含这些媒体类型，那么 responseXML 的值为 null。无论何时，只要 readyState 值不为 4，那么该 responseXML 的值也不为 null。

其实，这个 responseXML 属性值是一个文档接口类型的对象，用来描述被分析的文档。如果文档不能被分析（例如，如果文档不是良构的或不支持文档相应的字符编码），那么 responseXML 的值将为 null。

● status 属性

status 属性描述了 HTTP 状态代码，而且其类型为 short。仅当 readyState 值为 3（正在接收中）或 4（已加载）时，这个 status 属性才可用。当 readyState 的值小于 3 时，试图存取 status 的值将引发一个异常。

● statusText 属性

statusText 属性描述了 HTTP 状态代码文本；并且仅当 readyState 值为 3 或 4 才可用。当 readyState 为其他值时，试图存取 statusText 属性将引发一个异常。

3. 创建 XMLHttpRequest 对象的方法

XMLHttpRequest 对象提供了各种方法用于初始化和处理 HTTP 请求，下面对此进行详细介绍。

● abort()方法

用户可以使用 abort()方法来暂停与一个 XMLHttpRequest 对象相联系的 HTTP 请求，从而把该对象复位到未初始化状态。

● open()方法

用户需要调用 open()方法来初始化一个 XMLHttpRequest 对象。其中，method 参数是必须提供的，用于指定你想用来发送请求的 HTTP 方法。为了把数据发送到服务器，应该使用 POST 方法；为了从服务器端检索数据，应该使用 GET 方法。

● send()方法

在通过调用 open()方法准备好一个请求之后，用户需要把该请求发送到服务器。仅当 readyState 值为 1 时，用户才可以调用 send()方法；否则，XMLHttpRequest 对象将引发一个异常。

● setRequestHeader()方法

setRequestHeader()方法用来设置请求的头部信息。当 readyState 值为 1 时，用户可以在调用 open()方法后调用这个方法；否则，将得到一个异常。

● getResponseHeader()方法

getResponseHeader()方法用于检索响应的头部值。仅当 readyState 值是 3 或 4（换句话说，在响应头部可用以后）时，才可以调用这个方法；否则，该方法返回一个空字符串。

● getAllResponseHeaders()方法

getAllResponseHeaders()方法以一个字符串的形式返回所有的响应头部（每一个头部占单

独的一行）。如果 readyState 的值不是 3 或 4，则该方法返回 null。

22.2.2　发出 Ajax 请求

在 Ajax 中，许多使用 XMLHttpRequest 的请求都是从一个 HTML 事件（例如一个调用 JavaScript 函数的按钮点击事件或一个按键事件）中被初始化的。Ajax 支持包括表单校验在内的各种应用程序。有时，在填充表单的其他内容之前要求校验一个唯一的表单域。例如，要求使用一个唯一的 UserID 来注册表单。如果不是使用 Ajax 技术来校验这个 UserID 域，那么整个表单都必须被填充和提交。如果该 UserID 不是有效的，这个表单必须被重新提交。例如，相应于一个要求必须在服务器端进行校验的 Catalog ID 的表单域可按下列形式指定：

```
<form name="validationForm" action="validateForm" method="post">
<table>
 <tr><td>Catalog Id:</td>
   <td>
     < input type="text" size="20" id="catalogId" name="catalogId" autocomplete="off" onkeyup=
"sendRequest()">
   </td>
   <td><div id="validationMessage"></div></td>
 </tr>
</table></form>
```

前面的 HTML 使用 validationMessage div 来显示对应于这个输入域 Catalog Id 的一个校验消息。onkeyup 事件调用 JavaScript 的 sendRequest()函数。这个 sendRequest()函数创建一个 XMLHttpRequest 对象。创建 XMLHttpRequest 对象的过程因浏览器实现的不同而有所区别。

如果浏览器支持 XMLHttpRequest 对象作为一个窗口属性，那么，代码可以调用 XMLHttpRequest 的构造器。如果浏览器把 XMLHttpRequest 对象实现为一个 ActiveXObject 对象，那么，代码可以使用 ActiveXObject 的构造器。下面的代码将调用一个 init()函数。

```
<script type="text/javascript">
function sendRequest(){
 var xmlHttpReq=init();
 function init(){
   if (window.XMLHttpRequest) {
     return new XMLHttpRequest();
   }
 else if (window.ActiveXObject) {
   return new ActiveXObject("Microsoft.XMLHTTP");
 }
 }
</script>
```

接下来，用户需要使用 Open()方法初始化 XMLHttpRequest 对象，从而指定 HTTP 方法和

要使用的服务器 URL。

```
var catalogId=encodeURIComponent(document.getElementById("catalogId").value);
xmlHttpReq.open("GET",  "validateForm?catalogId=" + catalogId,  true);
```

默认情况下，使用 XMLHttpRequest 发送的 HTTP 请求是异步进行的，但是用户可以显式地把 async 参数设置为 true。在这种情况下，对 URL validateForm 的调用将激活服务器端的一个 servlet。但是用户应该能够注意到服务器端技术不是根本性的；实际上，该 URL 可能是一个 ASP、ASP.NET 或 PHP 页面或者一个 Web 服务，只要该页面能够返回一个响应，指示 CatalogID 值是否是有效的即可。因为用户在进行异步调用时，需要注册一个 XMLHttpRequest 对象来调用回调事件处理器，当它的 readyState 值改变时调用。记住，readyState 值的改变将会激发一个 readystatechange 事件。这时可以使用 onreadystatechange 属性来注册该回调事件处理器。

```
xmlHttpReq.onreadystatechange=processRequest;
```

然后，需要使用 send()方法发送该请求。因为这个请求使用的是 HTTP GET 方法，所以，用户可以在不指定参数或使用 null 参数的情况下调用 send()方法。

```
xmlHttpReq.send(null);
```

22.2.3　处理服务器响应

在上述示例中，因为 HTTP 方法是 GET，所以在服务器端接收的 servlet 将调用一个 doGet()方法，该方法将检索在 URL 中指定的 catalogId 参数值，并且从一个数据库中检查它的有效性。

该示例中的 servlet 需要构造一个发送到客户端的响应；而且，这个示例返回的是 XML 类型，因此，它把响应的 HTTP 内容类型设置为 text/xml 并且把 Cache-Control 头部设置为 no-cache。设置 Cache-Control 头部可以阻止浏览器简单地从缓存中重载页面。具体的代码如下。

```
public void doGet(HttpServletRequest request,
HttpServletResponse response)
throws ServletException,  IOException {
...
...
response.setContentType("text/xml");
response.setHeader("Cache-Control",  "no-cache");
}
```

从上述代码中可以看出，来自于服务器端的响应是一个 XML DOM 对象，此对象将创建一个 XML 字符串，其中包含要在客户端进行处理的指令。另外，该 XML 字符串必须有一个根元素。代码如下。

```
out.println("＜catalogId＞valid＜/catalogId＞");
```



XMLHttpRequest 对象设计的目的是处理由普通文本或 XML 组成的响应；但是，一个响应也可能是另外一种类型，如果用户代理支持这种内容类型的话。

当请求状态改变时，XMLHttpRequest 对象调用使用 onreadystatechange 注册的事件处理器。因此，在处理该响应之前，用户的事件处理器应该首先检查 readyState 的值和 HTTP 状态。当请求完成加载（readyState 值为 4）并且响应已经完成（HTTP 状态为 OK）时，用户就可以调用一个 JavaScript 函数来处理该响应内容。下列脚本负责在响应完成时检查相应的值并调用一个 processResponse()方法。

```
function processRequest(){
    if(xmlHttpReq.readyState==4){
        if(xmlHttpReq.status==200){
            processResponse();
        }
    }
}
```

该 processResponse()方法使用 XMLHttpRequest 对象的 responseXML 和 responseText 属性来检索 HTTP 响应。如上面所解释的，仅当响应的媒体类型是 text/xml、application/xml 或以+xml 结尾时，这个 responseXML 才可用。这个 responseText 属性将以普通文本形式返回响应。对于一个 XML 响应，用户将按如下方式检索内容，代码如下。

```
var msg=xmlHttpReq.responseXML;
```

借助于存储在 msg 变量中的 XML，用户可以使用 DOM 方法 getElementsByTagName()来检索该元素的值，代码如下。

```
var catalogId=msg.getElementsByTagName("catalogId")[0].firstChild.nodeValue;
```

最后，通过更新 Web 页面的 validationMessage div 中的 HTML 内容并借助 innerHTML 属性，用户可以测试该元素值以创建一个要显示的消息，代码如下：

```
if(catalogId=="valid"){
  var validationMessage = document.getElementById("validationMessage");
  validationMessage.innerHTML = "Catalog Id is Valid";
}
else
{
  var validationMessage = document.getElementById("validationMessage");
  validationMessage.innerHTML = "Catalog Id is not Valid";
}
```

22.3 CSS 与 Ajax 综合实战——制作自由拖动的网页

Ajax 综合了各个方面的技术，不但能够加快用户的访问速度，还可以实现各种特效。下面就制作一个自由拖放的网页，来巩固 CSS 与 Ajax 的综合使用。具体的操作步骤如下。

01 在 HTML 页面中创建用于存放数据的表格，代码如下。

```
<html>
<head>
<title>能够自由拖动布局区域的网页</title>
</head>
<body>
<table cellspacing="4" width="100%" id="parentTable">
<tr>
    <td width="25%" valgin="top">
        <table class="dragTable" cellspacing="0">
            <tr><td>蜂蜜</td></tr>
            <tr><td>蜂蜜，是昆虫蜜蜂从开花植物的花中采得的花蜜在蜂巢中酿制的蜜。蜜蜂从植物的花中采取含水量约为 80%的花蜜或分泌物，存入自己第二个胃中，在体内转化酶的作用下经过 30 分钟的发酵，回到蜂巢中吐出，蜂巢内温度经常保持在 35℃左右，经过一段时间，水份蒸发，成为水分含量少于 20%的蜂蜜，存贮到巢洞中，用蜂蜡密封。</td><tr>
        </table>
        <table class="dragTable" cellspacing="0">
            <tr><td>蜂王浆</td></tr>
            <tr><td>蜂王浆（royal jelly），又名蜂皇浆、蜂乳，蜂王乳，是蜜蜂巢中培育幼虫的青年工蜂咽头腺的分泌物，是供给将要变成蜂王的幼虫的食物。蜂王浆是高蛋白，并含有维生素 B 类和乙酰胆碱等。蜂王浆不能用开水或茶水冲服，并应该低温贮存。</td><tr>
        </table>
    </td>
    <td width="25%">
        <table class="dragTable" cellspacing="0">
            <tr><td>蜂花粉</td></tr>
            <tr><td>蜂花粉是有花植物雄蕊中的雄性生殖细胞，它不仅携带着生命的遗传信息，而且包含着孕育新生命所必需的全部营养物质，是植物传宗接代的根本，热能的源泉。蜂花粉是由蜜蜂从植物花中采集的花粉经蜜蜂加工成的花粉团，被誉为“全能的营养食品”、“浓缩的天然药库”、“全能的营养库”、“内服的化妆品”、“浓缩的氨基酸”等，是“人类天然食品中的瑰宝”。</td><tr>
        </table>
    </td>
    <td width="25%">
        <table class="dragTable" cellspacing="0">
            <tr><td>蜂毒</td></tr>
            <tr><td>蜂毒是一种透明液体，具有特殊的芳香气味，味苦、呈酸性反应，pH 为 5.0～
```

```
5.5，比重为 1.1313。在常温下很快就挥发干燥至原来液体重量的 30%～40%，</td><tr>
                </table>
                <table class="dragTable" cellspacing="0">
                    <tr><td>蜂胶</td></tr>
                    <tr><td>蜂胶是蜜蜂从植物芽孢或树干上采集的树脂（树胶），混入蜜蜂口器中腺体的分泌物，再和花粉、蜂蜡加工制成的一种胶状物质，是蜂巢的保护伞。一个 5～6 万只的蜂群一年只能生产蜂胶 100～150 克，被誉为“紫色黄金”</td><tr>
                </table>
            </td>
        </tr>
    </table>
    </body>
    </html>
```

使用 IE 9.0 浏览器浏览的效果如图 22-4 所示。

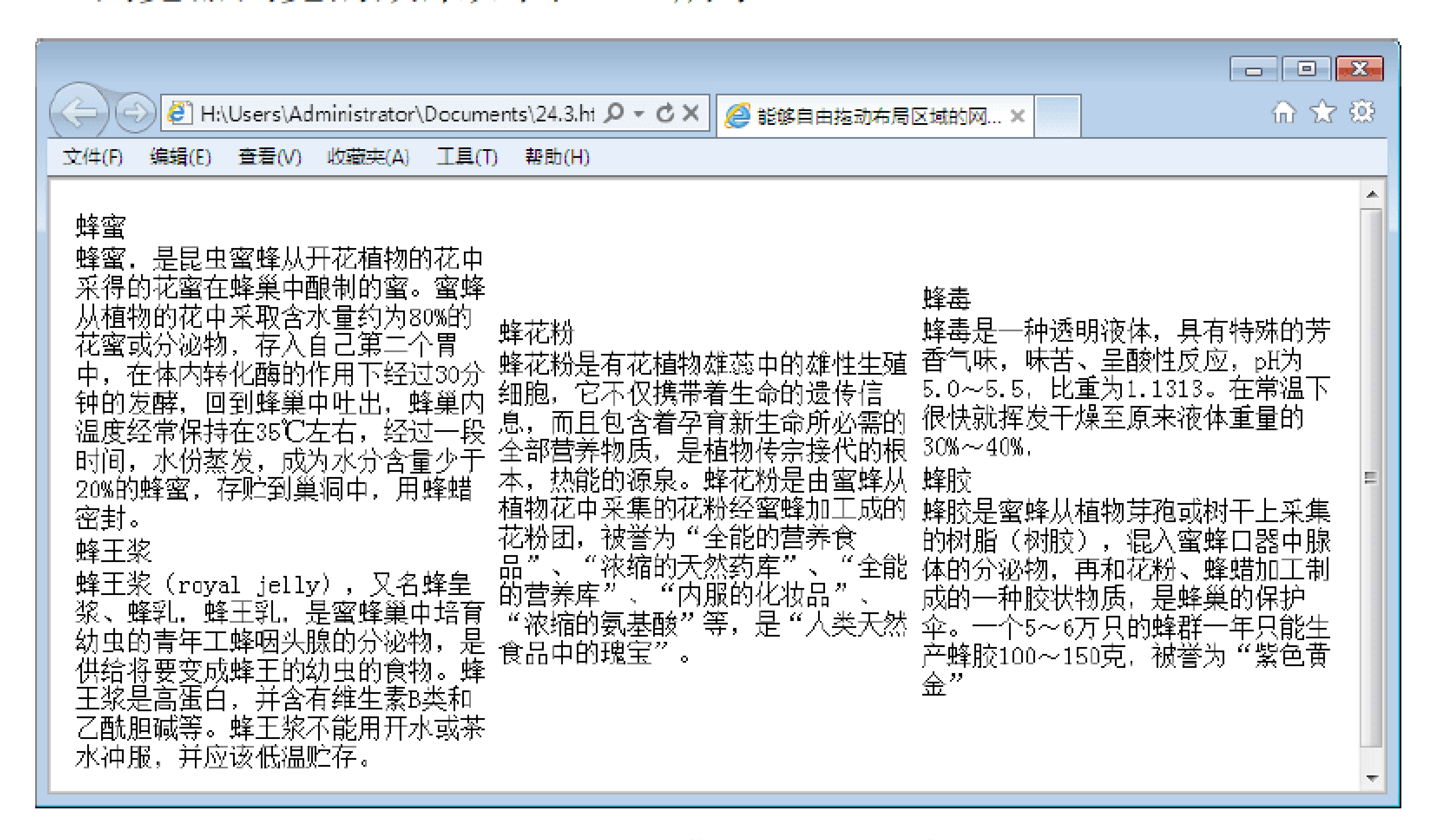


图 22-4　制作基本页面表格

02 为页面添加 Ajax 的 JavaScript 代码，以及 CSS 样式控制，使各个功能模块能自由拖放。

```
<style type="text/css">
<!--
body{
    font-size:12px;
    font-family:Arial, Helvetica, sans-serif;
    margin:0px; padding:0px;
    /*background-color:#ffffd5;*/
    background-color:#e6ffda;
}
.dragTable{
    font-size:12px;
```

```
        /*border:1px solid #003a82;*/
        border:1px solid #206100;
        margin-bottom:5px;
        width:100%;
        /*background-color:#cfe5ff;*/
        background-color:#c9ffaf;
}
td{
        padding:3px 2px 3px 2px;
        vertical-align:top;
}
.dragTR{
        cursor:move;
        /*color:#FFFFFF;
        background-color:#0073ff;*/
        color:#ffff00;
        background-color:#3cb500;
        height:20px;
        font-weight:bold;
        font-size:14px;
        font-family:Arial, Helvetica, sans-serif;
}
#parentTable{
        border-collapse:collapse;
}
-->
</style>
<script language="javascript" defer="defer">
var Drag={
        dragged:false,
        ao:null,
        tdiv:null,
        dragStart:function(){
                Drag.ao=event.srcElement;
                if((Drag.ao.tagName=="TD")||(Drag.ao.tagName=="TR")){
                        Drag.ao=Drag.ao.offsetParent;
                        Drag.ao.style.zIndex=100;
                  }else
                        return;
                Drag.dragged=true;
                Drag.tdiv=document.createElement("div");
                Drag.tdiv.innerHTML=Drag.ao.outerHTML;
```

```
            Drag.ao.style.border="1px dashed red";
            Drag.tdiv.style.display="block";
            Drag.tdiv.style.position="absolute";
            Drag.tdiv.style.filter="alpha(opacity=70)";
            Drag.tdiv.style.cursor="move";
            Drag.tdiv.style.border="1px solid #000000";
            Drag.tdiv.style.width=Drag.ao.offsetWidth;
            Drag.tdiv.style.height=Drag.ao.offsetHeight;
            Drag.tdiv.style.top=Drag.getInfo(Drag.ao).top;
            Drag.tdiv.style.left=Drag.getInfo(Drag.ao).left;
            document.body.appendChild(Drag.tdiv);
            Drag.lastX=event.clientX;
            Drag.lastY=event.clientY;
            Drag.lastLeft=Drag.tdiv.style.left;
            Drag.lastTop=Drag.tdiv.style.top;
        },
        draging:function(){//判断 MOUSE 的位置
            if(!Drag.dragged||Drag.ao==null)return;
            var tX=event.clientX;
            var tY=event.clientY;
            Drag.tdiv.style.left=parseInt(Drag.lastLeft)+tX-Drag.lastX;
            Drag.tdiv.style.top=parseInt(Drag.lastTop)+tY-Drag.lastY;
            for(var i=0;i<parentTable.cells.length;i++){
                var parentCell=Drag.getInfo(parentTable.cells[i]);
                if(tX>=parentCell.left&&tX<=parentCell.right&&tY>=parentCell.top&&tY<=parentCell.bottom){
                    var subTables=parentTable.cells[i].getElementsByTagName("table");
                    if(subTables.length==0){
                    if(tX>=parentCell.left&&tX<=parentCell.right&&tY>=parentCell.top&&tY<=
                    parentCell.bottom){
                            parentTable.cells[i].appendChild(Drag.ao);
                        }
                        break;
                    }
                    for(var j=0;j<subTables.length;j++){
                        var subTable=Drag.getInfo(subTables[j]);
                        if(tX>=subTable.left&&tX<=subTable.right&&tY>=subTable.top&&tY<=
                        subTable.bottom){
                            parentTable.cells[i].insertBefore(Drag.ao,subTables[j]);
                            break;
                        }else{
                            parentTable.cells[i].appendChild(Drag.ao);
                        }
```

```
                }
            }
        }
    },
    dragEnd:function(){
        if(!Drag.dragged)
            return;
        Drag.dragged=false;
        Drag.mm=Drag.repos(150,15);
        Drag.ao.style.borderWidth="0px";
        //Drag.ao.style.border="1px solid #003a82";
        Drag.ao.style.border="1px solid #206100";
        Drag.tdiv.style.borderWidth="0px";
        Drag.ao.style.zIndex=1;
    },
    getInfo:function(o){//取得坐标
        var to=new Object();
        to.left=to.right=to.top=to.bottom=0;
        var twidth=o.offsetWidth;
        var theight=o.offsetHeight;
        while(o!=document.body){
            to.left+=o.offsetLeft;
            to.top+=o.offsetTop;
            o=o.offsetParent;
        }
        to.right=to.left+twidth;
        to.bottom=to.top+theight;
        return to;
    },
    repos:function(aa,ab){
        var f=Drag.tdiv.filters.alpha.opacity;
        var tl=parseInt(Drag.getInfo(Drag.tdiv).left);
        var tt=parseInt(Drag.getInfo(Drag.tdiv).top);
        var kl=(tl-Drag.getInfo(Drag.ao).left)/ab;
        var kt=(tt-Drag.getInfo(Drag.ao).top)/ab;
        var kf=f/ab;
        return setInterval(function(){
            if(ab<1){
                clearInterval(Drag.mm);
                Drag.tdiv.removeNode(true);
                Drag.ao=null;
                return;
```

```
                    }
                    ab--;
                    tl-=kl;
                    tt-=kt;
                    f-=kf;
                    Drag.tdiv.style.left=parseInt(tl)+"px";
                    Drag.tdiv.style.top=parseInt(tt)+"px";
                    Drag.tdiv.filters.alpha.opacity=f;
                }
                ,aa/ab)
        },
        inint:function(){
                for(var i=0;i<parentTable.cells.length;i++){
                        var subTables=parentTable.cells[i].getElementsByTagName("table");
                        for(var j=0;j<subTables.length;j++){
                                if(subTables[j].className!="dragTable")
                                        break;
                                subTables[j].rows[0].className="dragTR";
                                subTables[j].rows[0].attachEvent("onmousedown",Drag.dragStart);
                        }
                }
                document.onmousemove=Drag.draging;
                document.onmouseup=Drag.dragEnd;
        }
}
Drag.inint();
</script>
```

使用 IE 9.0 浏览器浏览的效果如图 22-5 所示。

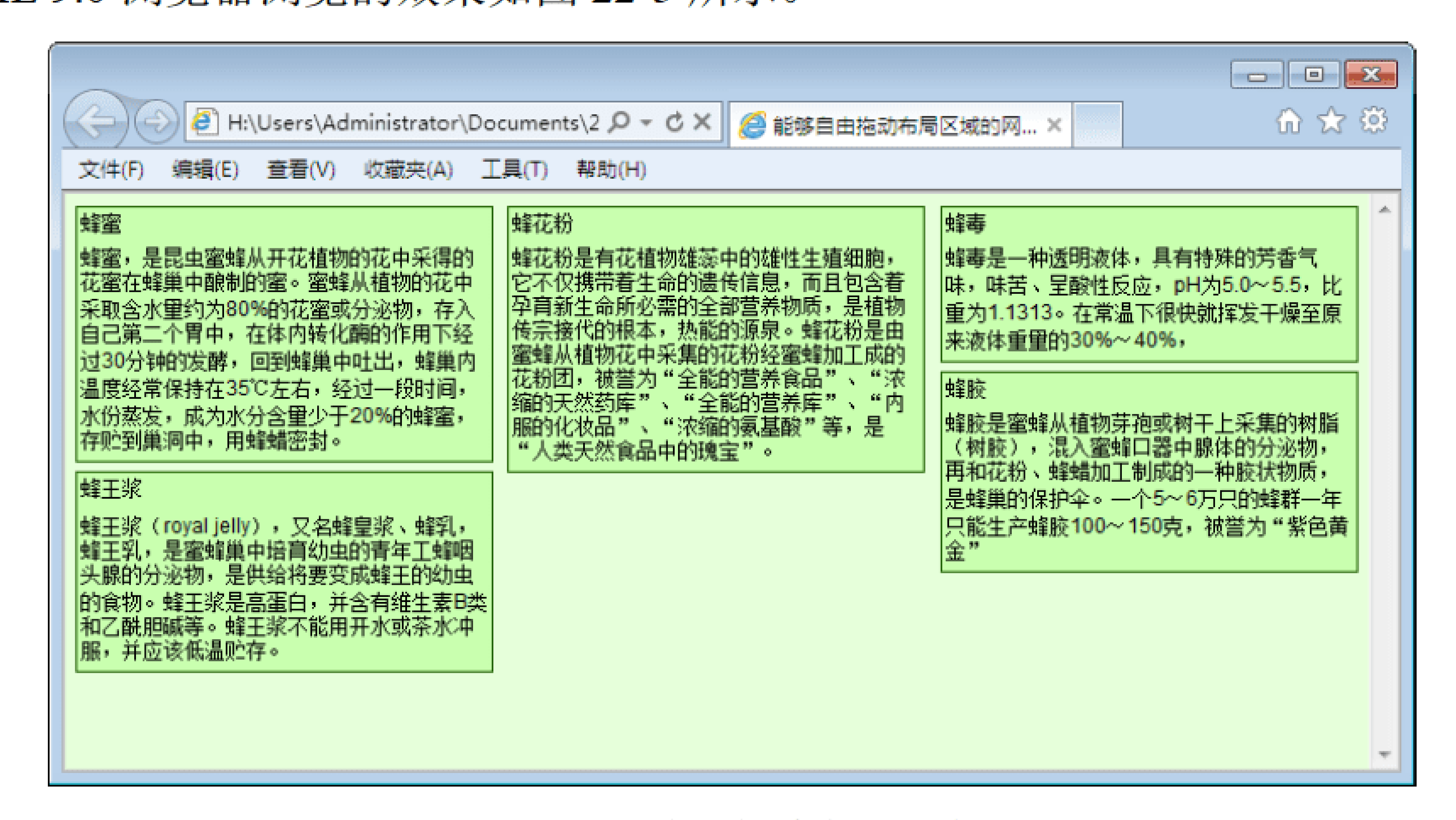


图 22-5　自由拖放布局区域

22.4　专家解惑

1．在发送 Ajax 请求时，是使用 GET 还是 POST？

与 POST 相比，GET 更简单也更快，并且在大部分情况下都能用。然而，在以下情况中，请使用 POST 请求：

- 无法使用缓存文件（更新服务器上的文件或数据库）。
- 向服务器发送大量数据（POST 没有数据量限制）。
- 发送包含未知字符的用户输入时，POST 比 GET 更稳定、更可靠。

2．在指定 Ajax 的异步参数时，应将该参数设置为 True 还是 False？

Ajax 指的是异步 JavaScript 和 XML（Asynchronous JavaScript and XML）。XMLHttpRequest 对象如果要用于 Ajax，其 open()方法的 async 参数必须设置为 true，代码如下：

```
xmlhttp.open("GET","ajax_test.asp",true);
```

对于 Web 开发人员来说，发送异步请求是一个巨大的进步。很多在服务器端执行的任务都相当费时。Ajax 出现之前，可能会引起应用程序挂起或停止。通过 AJAX，JavaScript 无需等待服务器的响应，而是在等待服务器响应时执行其他脚本，当响应就绪后对响应进行处理。

第 23 章

构建企业门户网站

一个公司的网站是这个公司形象的体现，其中也包含着整个公司的理念和方向。这种网站的特点是简约而不简单，没有过多的颜色修饰，大气的整体风格是最重要的。

23.1 构思布局

本实例是模拟一个小型软件公司的网站，公司主要承接电信方面的各种软件项目。网站上包括首页、产品信息、客户信息和联系我们等栏目。页面中采用红色和白色，红色部分显示导航菜单，白色部分显示文本信息。在 IE 9.0 中浏览效果如图 23-1 所示。



图 23-1 计算机网站首页

23.1.1 设计分析

作为一个软件公司的网站首页，其页面应简单、明了，给人以清晰的感觉。页头部分主要放置导航菜单和公司 Logo 信息等，其中 Logo 可以是一张图片或者文本信息；页面主体分为两个部分，页面主体左侧是公司介绍，对于公司介绍可以在首页上进行概括性描述；页面主体

右侧是新闻、产品和客户信息等，其中产品和客户的链接信息以列表形式显示，也可以通过页面顶部导航菜单进入相应的介绍页面。

对于网站的其他子页面，篇幅可以比较短，重点介绍软件公司的业务、联系方式、产品信息等，页面需要与首页风格相同。

23.1.2 排版架构

从图 23-1 可以看出，页面采用的是上中下结构，页面主体部分又嵌套了一个左右版式结构。其大致结构如图 23-2 所示。

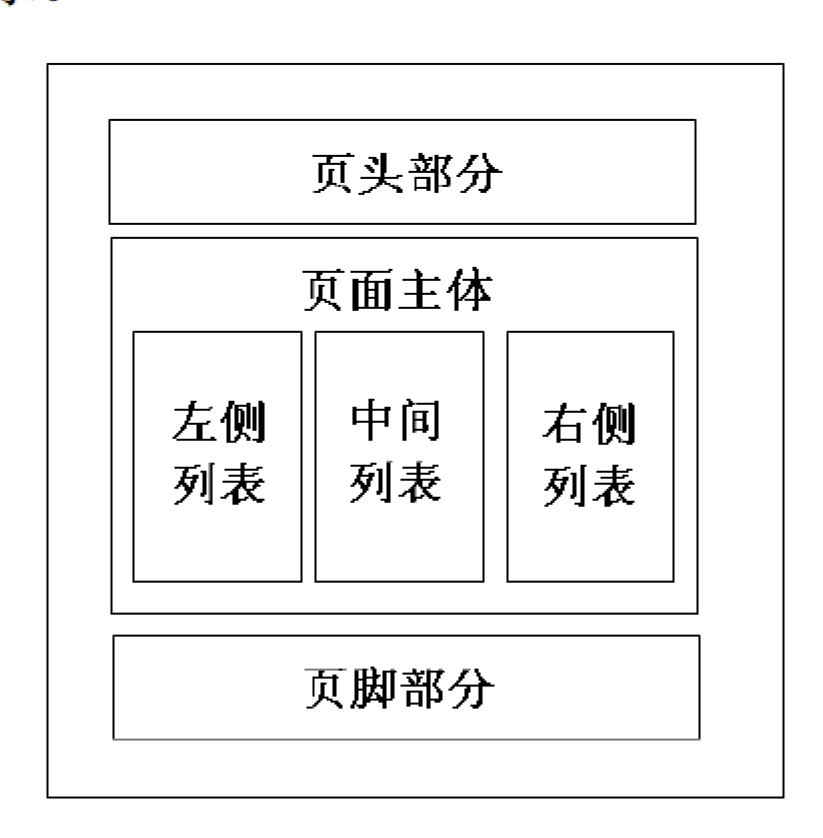


图 23-2　页面总体框架

在 HTML 页面中，通常使用 DIV 层对应上面不同的区域，可以是一个 DIV 层对应一个区域，也可以是多个 DIV 层对应同一个区域。本实例的 DIV 代码如下所示。

```
<div id="container">/*页面布局容器*/
<div id="top">
</div><!--end top-->
<div id="header">
</div><!--end header-->
<div id=me>/*导航菜单*/
</div>
<div id="content">
<div id="text">/*页面主体左侧内容*/
</div><!--end text-->
<div id="column">/*页面主体右侧内容*/
</div><!--end column-->
</div><!--end content-->
<div id="footer">/*页脚部分*/
</div><!--end footer-->
</div><!--end container-->
```

在上面的代码中，ID 名称为 container 的层是整个页面的布局容器，层 top、层 header 和

层 me 共同组成了页头部分，层 top 用于显示页面 logo，层 header 用于显示页头文本信息，层 me 用于显示页头导航菜单信息。页面主体是 content 层，其包含了两个层，text 层和 column 层，text 层是页面主体左侧内容，显示公司介绍信息；column 层是页面主体右侧内容，显示公司常用的导航链接。层 footer 是页脚部分，用于显示版权信息和地址信息。

在 CSS 文件中，层 container 和层 content 对应的 CSS 代码如下所示。

```
#container
{
 margin: 0pt auto;
 width: 770px;
 position: relative; }
#content {
 background: transparent url('images/content.gif') repeat-y;
 clear: both;
 margin-top: 5px;
 width: 770px;
}
```

在上面的代码中，#container 选择器定义了整个布局容器的宽度、外边距和定位方式。#content 选择器定义了背景图片、宽度和顶部边距。

23.2　模块分割

当页面整体架构完成后，就可以动手制作不同的模块区域了。其制作流程采用自上而下，从左到右的顺序。然后再对页面样式进行整体调整。

23.2.1　Logo 与导航菜单

一般情况下，Logo 信息和导航菜单都放在页面顶部，作为页头部分。其中 Logo 信息作为公司标志，通常放在页面的左上角或右上角；导航菜单放在页头部分和页面主体两者之间，用于链接其他的页面。在 IE 9.0 中浏览效果如图 23-3 所示。



图 23-3　页面 Logo 和导航菜单

在 HTML 文件中，用于实现页头部分的 HTML 代码如下所示。

```
<div id="top">
</div><!--end top-->
<div id="header">
```

```
<h1>计算机 网站</h1>
</div><!--end header-->
<div id=me>
<ul id="menu">
<li><a href="#" class="actual">首页</a></li>
<li><a href="#" >产品</a></li>
<li><a href="#">客户</a></li>
<li><a href="#">联系方式</a></li>
</ul>
</div>
```

在上面的代码中，层 top 用于显示页面 logo，层 header 用于显示页头的文本信息，如公司名称；层 me 用于显示页头导航菜单。在层 me 中，有一个无序列表，用于制作导航菜单，每个选项都是由超级链接组成的。

在 CSS 样式文件中，对应上面标记的 CSS 代码如下所示。

```
#top {
 background: transparent url('images/top.jpg') no-repeat;
 height: 50px;
}
#top p {
 margin: 0pt;
 padding: 0pt;
}
#header {
 background: transparent url('images/header.jpg') no-repeat;
 height: 150px;
 margin-top: 5px;
}
#menu {
 position: absolute;
 top: 180px;
 left: 15px;
}
#header h1 {
 margin: 5px 0pt 0pt 50px;
 padding: 0pt;
 font-size: 1.7em;
 }
#header h2 {
  margin: 10px 0pt 0pt 90px;
 padding: 0pt;
```

```
  font-size: 1.2em;
  color: rgb(223, 139, 139);
}
ul#menu {
  margin: 0pt;
}
#menu li {
  list-style-type: none;
  float: left;
  text-align: center;
  width: 104px;
  margin-right: 3px;
  font-size: 1.05em;
}
#menu a {
  background: transparent url('images/menu.gif') no-repeat;
  overflow: hidden;
  display: block;
  height: 28px;
  padding-top: 3px;
  text-decoration: none;
  twidth: 100%;
  font-size: 1em;
  font-family: Verdana,"Geneva CE",lucida,sans-serif;
  color: rgb(255, 255, 255);
}
#menu li > a, #menu li > strong {
  width: auto;
}
#menu a.actual {
  background: transparent url('images/menu-actual.gif') no-repeat;
  color: rgb(149, 32, 32);
}
#menu a:hover {
  color: rgb(149, 32, 32);
}
```

在上面的代码中，#top 选择器定义了背景图片和层高；#header 定义了背景图片、高度和顶部外边距；#menu 层定义了层定位方式和坐标位置。其他选择器分别定义了上面 3 个层中元素的显示样式，如段落显示样式、标题显示样式和超级链接样式等。

23.2.2　左侧文本介绍

在页面主体中，其左侧部分主要介绍公司相关信息。左侧文本采用的是左浮动并且固定宽度的版式设计，重点在于调节宽度使不同浏览器之间能够效果一致，并且颜色与 Logo 和左侧的导航菜单相匹配，使整个网站和谐、大气。在 IE 9.0 中浏览效果如图 23-4 所示。



图 23-4　页面左侧文本介绍

在 HTML 文件中，创建页面左侧内容介绍的代码如下所示。

```
<div id="content">
<div id="text">
<h3 class="headlines"><a href="#" title="testing">欢迎来到我们的网站 </a></h3>
<p><img src="images/fotos.jpg" alt="fotos" align="right" />
远大公司成立于 1998 年，注册资本 1700 万元。是国家认定的高新技术企业、软件企业，是专业的电信系统仿软件和应用服务供应商。</p><p>
公司坚持走自立创新、稳步发展的道路，以创立品牌为自己的基本策略，以产品自身的品质，先进的技术和良好的服务取信于用户。2002 年至今公司先后有多个软件产品获得了河南省信息产业厅颁发的《软件产品登记证书》和国家版权局颁发的《软件著作权登记证书》。同时远大的进步和发展，也得到了政府部门的大力支持和关注，获得国家科技部和省、市政府部门技术创新基金无偿资助百余万元。并正式获得中国质量体系认证中心颁发的 ISO9001:2008 质量管理体系认证证书。
</p>
<p> </p>
</div><!--end text-->
</div>
```

在上面的代码中，层 content 是页面主体，层 text 是页面主体中左侧部分。层 text 包含标题和段落信息，段落中包含一张图片。

在 CSS 文件中，针对上面 HTML 标记的 CSS 代码如下所示。

```
#text {
  background: rgb(255, 255, 255) url('text-top.gif') no-repeat;
  width: 518px;
  color: rgb(0, 0, 0);
  float: left;
}
#text h1, #text h2, #text h3, #text h4 {
  color: rgb(140, 9, 9);
}
#text h3.headlines a {
  color: rgb(140, 9, 9);
}
```

在上面的代码中，#text 层定义了背景图片、背景颜色、字体颜色和页面左浮动。下面的选择器定义了标题显示样式，如字体颜色等。#text h3.headlines a 选择器定义了标题 3、类 headlines 和超级链接显示样式。

23.2.3　右侧导航链接

在页面主体右侧部分中，其文本信息不是太多，但非常重要，是首页用于链接其他页面的导航链接，如客户详细信息、最新消息等。同样右侧版式需要设置为固定宽度并且向左浮动的版式。在 IE 9.0 中浏览页面效果如图 23-5 所示。



图 23-5　页面右侧的超级链接

从图 23-5 可知，需要包含几个无序列表和标题，其中列表选项为超级链接。HTML 文件中用于创建页面主体右侧版式的代码，如下所示。

```
<div id="column">
<h3><span>最新消息</span></h3>
```

```
<ul class="category_list"><li><a href="#">公司组织员工连云港旅游</a></li>
<li><a href="#">2011 员工乒乓球大赛开幕</a></li>
<li><a href="#">公司总经理会见实习大学生</a></li>
<li><a href="#">公司销售部门再传捷报</a></li></ul>
<h3><span>产品展示</span></h3>
<ul class="recent_articles"><li><a href="#">在线人员素质考核系统</a></li>
<li><a href="#">线损计算机系统</a></li>
<li><a href="#">质量运用管理系统</a></li></ul>
<h3><span>客户</span></h3>
<ul class="wet_recent_comments"><li><a href="#"><cite>华中地区</cite></a><p>河南地区</p></li>
<li><a href="#"><cite>华东地区</cite></a><p>上海地区</p></li></ul>
</div><!--end column-->
<div id="content-bottom"> </div>
```

在上面的代码中创建了两个层，分别为 column 层和 content-bottom 层。其中 column 层用于显示页面主体中的右侧链接，并包含了 3 个标题和 3 个超级链接。content-bottom 层用于消除上一层的浮动效果。

在 CSS 文件中，用于修饰上面 HTML 标记的 CSS 代码如下所示。

```
#column {
  background: rgb(142, 14, 14) url('images/column.gif') no-repeat;
  float: right;
  width: 247px; }
#column p { font-size: 0.7em; }
#column ul { font-size: 0.8em; }
#column h3 {
  background: transparent url('images/h3-column.gif') no-repeat;
  position: relative;
  left: -18px;
  height: 26px;
  width: 215px;
  margin-top: 10px;
  padding-top: 6px;
  padding-left: 6px;
  font-size: 0.9em;
  z-index: 1;
  font-family: Verdana,"Geneva CE",lucida,sans-serif;
}
#column h3 span { margin-left: 10px; }
#column span.name {
  text-align: right;
  color: rgb(223, 58, 0);
```

```
  margin-right: 5px;
}
#column a { color: rgb(255, 255, 255); }
#column a:hover { color: rgb(80, 210, 122); }
p.comments {
  text-align: right;
  font-size: 0.8em;
  font-weight: bold;
  padding-right: 10px;
}
#content-bottom {
  background: transparent url('images/content-bottom.gif') no-repeat scroll left bottom;
  clear: both;
  display: block;
  width: 770px;
  height: 13px;
  font-size: 0pt;
}
```

在上面的代码中，#column 选择器定义背景图片、背景颜色、页面右浮动和宽度。#content-bottom 选择器定义背景图片、宽度、高度、字体大小和以块显示，并且使用 clear 消除前面层使用 float 的影响。其他选择器主要定义 column 层中其他元素的显示样式，如无序列表样式、列表选项样式和超级链接样式等。

23.2.4 版权信息

版权信息一般放置在页面底部，用于介绍页面的作者、地址信息等，是页脚的一部分。页脚部分和其他网页部分一样，需要保持简单、清晰的风格。在 IE 9.0 中浏览效果如图 23-6 所示。

网页设计者：李四工作室

图 23-6 页脚部分

从图 23-6 可以看出，页脚部分非常简单，只包含了一个作者信息的超级链接，因此设置起来比较方便，其代码如下。

```
<div id="footer">
<p id="ivorius"><a href="#">网页设计者：李四工作室</a></p>
</div><!--end footer-->
```

在上面的代码中，层 footer 包含了一个段落信息，其中段落的 id 是 ivorius。

在 CSS 文件中，用于修饰上面 HTML 标记的样式代码如下所示。

```
#footer {
```

```
    background: transparent url('images/footer.png') no-repeat scroll left bottom;
    margin-top: 5px;
    padding-top: 2px;
    height: 33px;
  }
  #footer p { text-align: center; }
  #footer a { color: rgb(255, 255, 255); }
  #footer a:hover { color: rgb(223, 58, 0); }
  p#ivorius {
    float: right;
    margin-right: 13px;
    font-size: 0.75em;
  }
  p#ivorius a { color: rgb(80, 210, 122); }
```

在上面的代码中，#footer 选择器定义了页脚背景图片、内外边距的顶部距离和高度。其他选择器定义了页脚部分文本信息的对齐方式、超级链接样式等。

23.3 整体调整

前面完成了首页中不同部分的制作，其整个页面基本上都已经成形。在制作完成后，需要根据页面实际效果做一些细节上的调整，从而完善页面整体效果。例如，各块之间的 padding 和 margin 值是否与页面整体协调，各个子块之间是否协调统一等。在页面调整前，在 IE 9.0 中浏览效果如图 23-7 所示。

从图 23-7 中，可以发现页面段落没有缩进，页面右侧列表选项之间距离太小。这时可以利用 CSS 属性调整，其代码如下所示。



图 23-7 页面调整前的效果

```
p { margin: 0.4em 0.5em; font-size: 0.85em;text-indent:2em; }
a { color: rgb(25, 126, 241); text-decoration: underline; }
a:hover { color: rgb(223, 58, 0); text-decoration: none; }
a img { border: medium none ; }
ul, ol { margin: 0.5em 2.5em; }
h2 { margin: 0.6em 0pt 0.4em 0.4em; }
h3, h4, h5 { margin: 1em 0pt 0.4em 0.4em; }
* { margin: 0pt; padding: 0pt; }
body { background: rgb(61, 62, 63) url('images/body.gif') repeat; color: white; font-size: 1em; font-family: "Trebuchet MS",Tahoma,"Geneva CE",lucida; }
```

在上面的代码中，全局选择器“*”设置了内外边距距离，body 标记选择器设置了背景颜色、图片、字体大小、字体颜色和字形等。其他选择器分别设置了段落、超级链接、标题和列表等样式信息。

23.4　专家解惑

1．Firefox 和 IE 浏览器如何处理负边界问题？

在 IE 中，超出父元素的部分会被父元素覆盖，而在 Firefox 中，超出父元素的部分会覆盖父元素，但前提是父元素有边框或内边距，不然负边距会显示在父元素上，使得父元素拥有负边距。在进行网页设计时，针对上面的情况可以对元素进行相对定位。

2．在定义子元素的上边距时，如果超出元素高度，应怎么处理？

在 IE 浏览器中，子元素上边距显示正常，而在 Firefox 浏览器中，子元素上边距显示在父元素上方。其解决办法是在父元素中增加 overflow:hidden 语句或给父元素增加边框或内边距。

第 24 章

团购商务类网站的设计

电子商务网站类型较多，本实例以当下比较流行的团购类网站为例进行介绍。团购这一名词是最近几年才出现的，有关团购的商业类网站也如雨后春笋般迅速火爆起来。比较有名的团购网站有聚划算、窝窝团、拉手网、美团网等。

24.1 整体布局

团购网根据薄利多销、量大价优的原理，将大量的散户聚集起来，共同购买一件或多件商品。这样，商家可以快速处理大批量的商品，低价销售，以量牟利。而对于购买者来说，可以以远远低于市场单品的价格购得满意的商品，有的甚至低于批发价格。这极大地促进了市场的交易能力。

本实例就来制作一个典型的团购类网页，最终效果如图 24-1 所示。



图 24-1 团购网页面效果

24.1.1 整体设计分析

团购网站已经成为典型，在设计这类网站时应当体现出以下几点。

- 涉及面广：团购网站本身的面向对象没有特殊的限制，任何具有一定消费群体的商品都可以出现在团购网站上，所以一个团购网站本身就应该涉及生活的方方面面，以方便浏览者访问所需要的类别。
- 考虑区域性：很多和生活较贴近的团购内容（如饮食），有明显的地区性限制，一般只能让当地人去团购，所以团购中应该有区域选择的模块。

- 展示最新商品：团购网站贴近生活，每天都有可能有新的商家发布团购信息，而这些信息往往都是浏览者比较关注的内容。所以要在页面的主体位置使用大块区域显示最新的团购信息，并配置详细的图文信息。
- 数量登记：团购网站之所以能够成团，并非随便几个人购买就可以，人数太少对商家来说有损失，因此团购商品一般都有人数限制，达到指定人数后才可成团。所以商品要有参团人数登记功能，并且将参团信息展示给新的浏览者。
- 友情帮助：团购网站不是简单的浏览型网站，而是具有网络电子商务功能的网站，所有的浏览者都需要按照电子商务平台的流程和规范进行操作，因此对于一些常规的操作事项要提供帮助连接。
- 格调温馨：团购网站面向的对象是普通大众，要提供的应该是值得信赖的优质团购信息和服务，所以团购网站的风格要温馨、大气。

每一家团购网站都有自己的特色。以上分析内容只是一般团购网站应当具备的基本功能。具体的个性化设计需要各自团购公司的创意。而本实例就依照上述的基本功能来制作一个简单的团购商业类网页。

24.1.2 架构布局分析

本实例网站采用的是典型的上中下结构，中间又可以分为左右结构。整体的排版架构如图 24-2 所示。

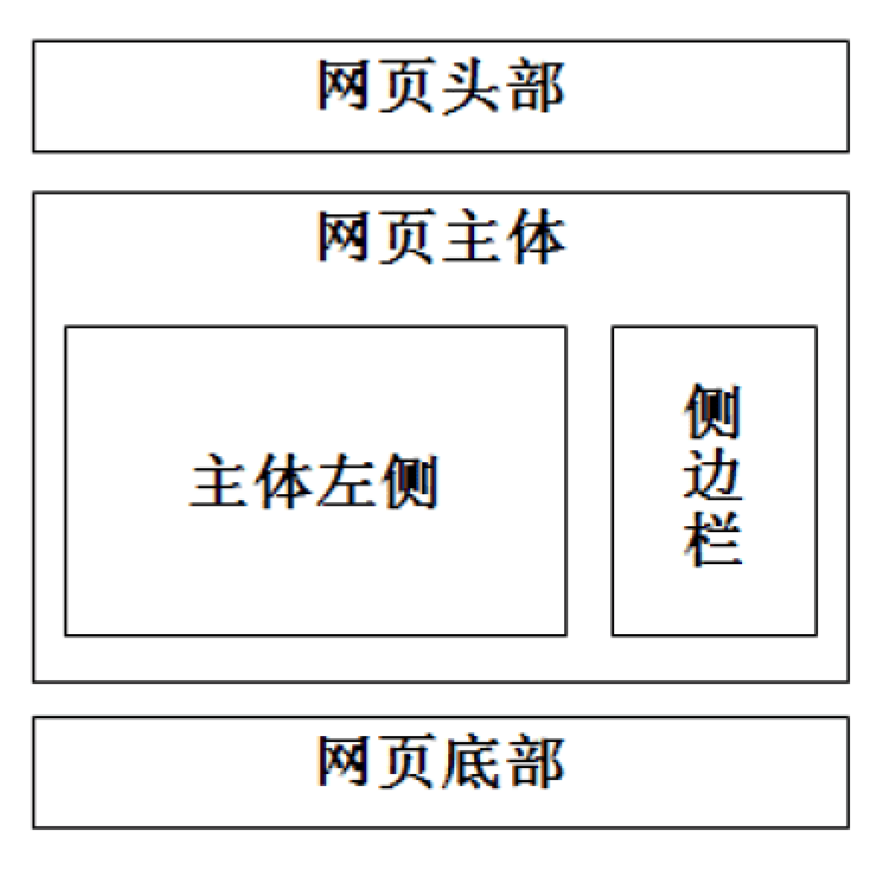


图 24-2 网页布局

电子商务类网站所要实现的功能较多，模块组成相对也比较多。依照网页的上中下结构，模块组成如下。

1. 网页头部

网页头部包括网页 logo 模块、信息搜索模块和导航菜单栏模块。最终效果如图 24-3 所示。



图 24-3　网页头部

2. 网页主体

网页主体内容较多，主要包括团购分类及区域选择模块、最新团购商品展示模块、热门分类模块、热销商品排行模块和热门城市索引模块，如图 24-4 所示。

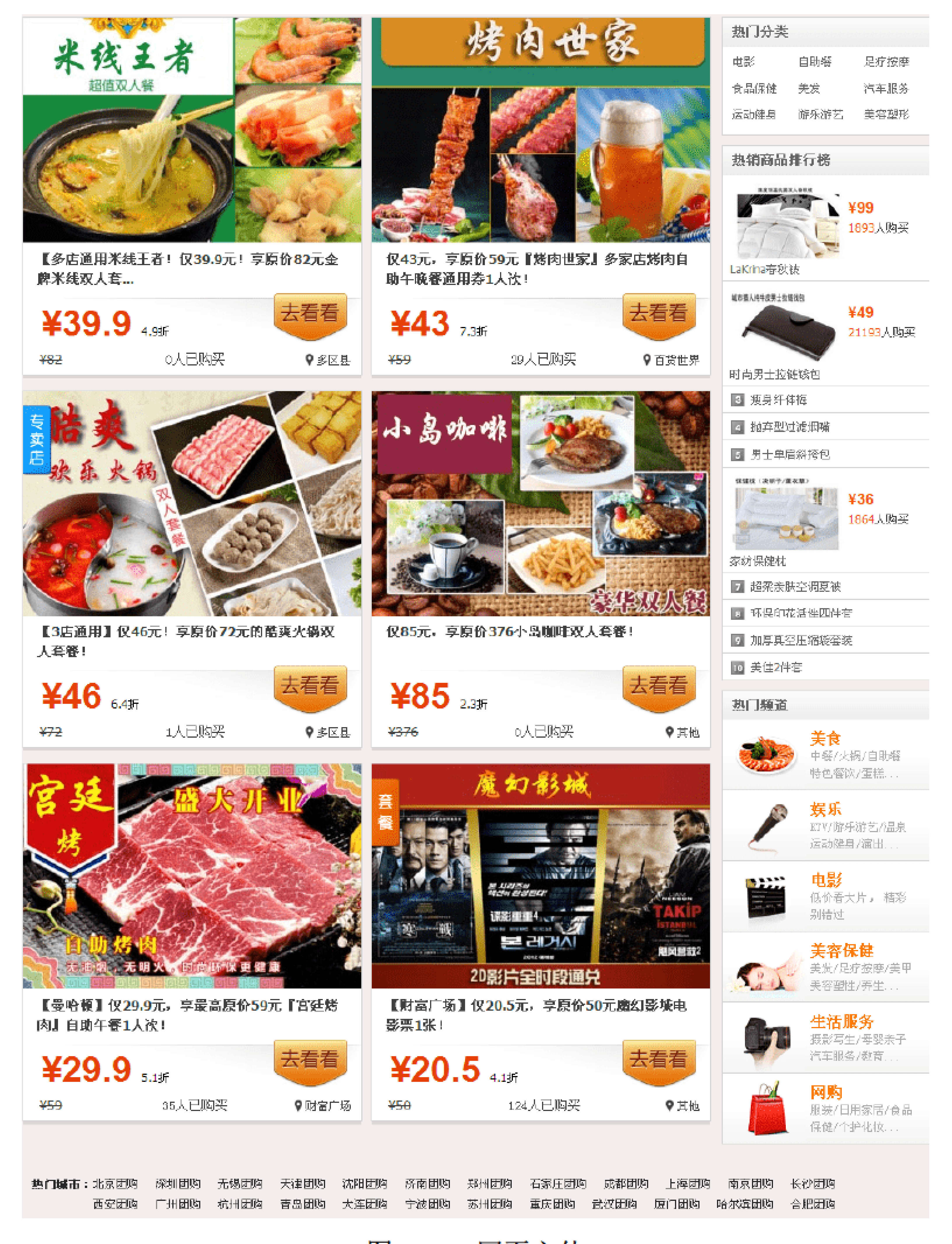


图 24-4　网页主体

3. 网页底部

底部主要是客户服务和快捷链接模块，用于解决各种客户服务问题，如图 24-5 所示。

用户帮助
玩转阿里
常见问题
秒杀规则
积分规则
消费者保障
网站地图

获取更新
阿里团新浪微博
阿里团开心网主页
阿里团豆瓣小组
RSS订阅
手机版下载

商务合作
商家入驻
提供团购信息
友情链接
开放API

公司信息
关于我们
媒体报道
加入我们
隐私声明
用户协议

24小时服务热线
500-000-0000
500-000-0000
我要提问

图 24-5　网页底部

主要使用 DIV 来实现各个模块的分割，其代码如下。

```
<div  id=headmin> </div>
//网页头部
<div  id=headnav> </div>
//导航菜单栏
<div  class="dhnav_box  dhnav_box_index  clearfix  "> </div>
//网页主体——团购分类及区域选择
<div  class=con_boxindex> </div>
//网页主体——最新团购商品展示及右侧边栏
<div  class=hot_city> </div>
//热门城市链接
<div  id=footer> </div>
//网页底部
```

24.2　主要模块设计

网站制作要逐步完成，本实例中网页制作主要包括 7 个部分，详细制作方法如下。

24.2.1　样式代码分析

首先，网页设计中需要使用 CSS 样式表控制整体样式，所以网站可以使用以下代码结构实现页面代码框架和 CSS 样式的插入。

```
<html><head><title>阿里团</title>
<meta name=keywords content="阿里团">
<link rel=stylesheet type=text/css href="css/css.css">
<script type=text/javascript src="js/user.js"></script>
</head>
<body>
```

```
……
</body>
</html>
```

由以上代码可以看出，案例中使用了一个 CSS 样式表 css.css。其中包含网页通用样式及特定内容的样式。样式表内容如下。

```
html {
    font-family: tahoma, verdana, arial, sans-serif, "宋体"; background: #f3eded; color: #000
}
body {
    background: #f3eded
}
body {
    padding-bottom: 0px; margin: 0px; padding-left: 0px; padding-right: 0px; padding-top: 0px
}
div {
    padding-bottom: 0px; margin: 0px; padding-left: 0px; padding-right: 0px; padding-top: 0px
}
dl {
    padding-bottom: 0px; margin: 0px; padding-left: 0px; padding-right: 0px; padding-top: 0px
}
……
<!--================中间内容省略=================-->
……
#sp_nav_list .fenlei .sec_ul {
 border-bottom: #eeeeee 1px solid; border-left: #eeeeee 1px solid; padding-bottom: 0px; margin: 0px 0
px 10px 44px; padding-left: 0px; width: 880px; padding-right: 0px; background: #f8f8f8; border-top: #eeeeee
 1px solid; border-right: #eeeeee 1px solid; padding-top: 5px
}
#sp_nav_list .fenlei .sec_ul li {
 padding-bottom: 5px
}
#msglogin {
    display: none
}
```



本实例中的样式表比较多，这里只展示一部分，可参考下载资源中的代码。

24.2.2 网页顶部代码分析

网页头部包括网页 logo 模块、信息搜索模块、导航菜单栏模块。本实例中网页头部的效

果如图 24-6 所示。



图 24-6　网页头部

实现网页头部的详细代码如下所示。

```
<ul>
  <li class=seach>
  <div class=soso>
  <form id=soso_form method=post action=/g/search><input id=querystring
  value=搜商品、找商家、逛商圈 type=text name=querystring> <a id=soso_submit class=btu href="javas
cript:;">搜索</a> </form>
 <a href="#">帮助</a> </div></li>
  <li><a href="#"><img src="images/logo.gif"></a></li>
  <li class=title>
  <h1>精挑细选</h1></li>
  <li class=city>
  <h2 id=cityname>郑州</h2><span>【<a href="#" data-prarm="city_list">切换城市</a>】</span>
<div id=show_city class=bubble><b class=ico>ico</b>
 <b id=ipclose class=cloce>ico</b> 您是不是在<em id=ipcityname></em>？点击可选择其他城市 </di
v></li></ul></div>
<!--头部导航-->
<div id=headnav>
<ul id=nav>
  <li class=phone date-nav="pinpaihui"><a href="#" data-prarm="click_mobile_nav"><b>ico</b><span>
手机版</span>手机版</a></li>
  <li date-nav="shangcheng"><a href="#" data-prarm="click_ channel10"><b>ico</b>阿里商城</a> </li>
  <li date-nav="index"><a href="#" data-prarm="click_ channel1"><b>ico</b>团购精选</a> </li>
  <li date-nav="meishi"><a href="#" data-prarm="click_ channel2"><b>ico</b>美食</a> </li>
  <li date-nav="yule"><a href="#" data-prarm="click_ channel3"><b>ico</b>娱乐</a> </li>
  <li date-nav="dianying"><a href="#" data-prarm="click_ channel4"><b>ico</b>电影</a> </li>
  <li date-nav="meirongbaojian"><a href="#" data-prarm="click_channel5"><b>ico</b>美容保健</a></li>
  <li date-nav="shenghuofuwu"><a href="#" data-prarm="click_ channel6"><b>ico</b>生活服务</a></li>
  <li date-nav="lvyou"><a href="#" data-prarm="click_ channel7"><b>ico</b>旅行</a> </li>
  <li date-nav="jiudian"><a href="#" data-prarm="click_ channel8"><b>ico</b>酒店</a> </li>
  <li date-nav="shangpin"><a href="#" data-prarm="click_ channel9"><b>ico</b>网购</a> </li>
  <li date-nav="shop"><a href="#" data-prarm="click_ channel9"><b>ico</b>品牌汇</a><em class=new>
new</em>
</li></ul>
</div>
```

24.2.3　分类及区域选择模块分析

团购分类及区域选择模块在团购网站中是最常见的，本实例中该模块的效果图 24-7 所示。

分类： 团购精选 (100)　餐饮美食 (297)　休闲娱乐 (51)　电影 (7)　美容保健 (100)　生活服务 (178)　旅行 (1559)　酒店 (7254)　网购 (2941)
区县： 全部　金水区 (52)　二七区 (25)　管城区 (22)　中原区 (21)　郑东新区 (11)　上街区 (5)　惠济区 (9)　经济技术开发区 (6)　邙山区 (6)　高新开发区 (6)
出口加工区 (3)　巩义市 (3)　荥阳市 (3)　新密市 (3)　新郑市 (3)　登封市 (3)　中牟县 (3)　其他 (20)

图 24-7　分类及区域选择模块

该模块主要是文字和超链接，实现起来较简单，具体代码如下。

```
<div class="dhnav_box dhnav_box_index clearfix "><!--分类更多开始-->
  <div class="list_more_2 clearfix">
<ul class="pd_nav clearfix">
  <li class=lft>分类：</li>
  <li class=on date="all_btm"><a class="nav_lista1 clearfix" href="#" data-prarm="click_channel1_0">团购精选(100)</a> </li>
 <li date="all_btm"><a href="#" data-prarm="click_channel1_1">餐饮美食(<em>297</em>)</a></li>
 <li date="all_btm"><a href="#" data-prarm="click_channel1_2">休闲娱乐(<em>51</em>)</a> </li>
 <li date="all_btm"><a href="#" data-prarm="click_channel1_3">电影(<em>7</em>)</a> </li>
 <li date="all_btm"><a href="#" data-prarm="click_channel1_4">美容保健(<em>100</em>)</a></li>
 <li date="all_btm"><a href="#" data-prarm="click_channel1_5">生活服务(<em>178</em>)</a></li>
 <li date="all_btm"><a href="#" data-prarm="click_channel1_6">旅行(<em>1559</em>)</a> </li>
 <li date="all_btm"><a href="#" data-prarm="click_channel1_7">酒店(<em>7254</em>)</a> </li>
 <li date="all_btm"><a href="#" data-prarm="click_channel1_8">网购(<em>2941</em>)</a> </li></ul></div>
<!--区县更多开始-->
<div class="list_more_2 no_bottom clearfix">
<ul class="pd_nav clearfix">
  <li class=lft>区县：</li>
  <li class=on><a class="nav_lista1 clearfix"#" data-prarm="click_channel1_">全部</a> </li>
  <li><a href="#" data-prarm="click_channel1_jinshui">金水区(<em>52</em>)</a> </li>
  <li><a href="#" data-prarm="click_channel1_erqi">二七区(<em>25</em>)</a> </li>
  <li><a href="#" data-prarm="click_channel1_guancheng">管城区(<em>22</em>)</a> </li>
  <li><a href="#" data-prarm="click_channel1_zhongyuan">中原区(<em>21</em>)</a> </li>
  <li><a href="#" data-prarm="click_channel1_zhengdongxinqu">郑东新区(<em>11</em>)</a> </li>
  <li><a href="#" data-prarm="click_channel1_shangjie">上街区(<em>5</em>)</a> </li>
  <li><a href="#" data-prarm="click_channel1_huiji">惠济区(<em>9</em>)</a> </li>
  <li><a href="#" data-prarm="click_channel1_jishujingjikaifa">经济技术开发区(<em>6</em>)</a> </li>
  <li><a href="#" data-prarm="click_channel1_mangshan">邙山区(<em>6</em>)</a> </li>
  <li><a href="#" data-prarm="click_channel1_gaoxinkaifa">高新开发区(<em>6</em>)</a> </li>
  <li><a href="#" data-prarm="click_channel1_chukoujiagong">出口加工区(<em>3</em>)</a> </li>
  <li><a href="#" data-prarm="click_channel1_gongyi">巩义市(<em>3</em>)</a> </li>
```

```
    <li><a  href="#"  data-prarm="click_channel1_xingyang">荥阳市(<em>3</em>)</a>  </li>
    <li><a  href="#"  data-prarm="click_channel1_xinmi">新密市(<em>3</em>)</a>  </li>
    <li><a  href="#"  data-prarm="click_channel1_xinzheng">新郑市(<em>3</em>)</a>  </li>
    <li><a  href="#"  data-prarm="click_channel1_dengfeng">登封市(<em>3</em>)</a>  </li>
    <li><a  href="#"  data-prarm="click_channel1_zhongmu">中牟县(<em>3</em>)</a>  </li>
    <li><a  href="#"  data-prarm="click_channel1_other">其他(<em>20</em>)</a>  </li></ul></div></div>
```

24.2.4　新品展示模块分析

网页主体左侧为整个网页的主要内容，是最新团购产品的展示模块，该模块中使用大量醒目的图文展示产品信息，并且有价格和团购数量统计功能。具体效果如图 24-8 所示。

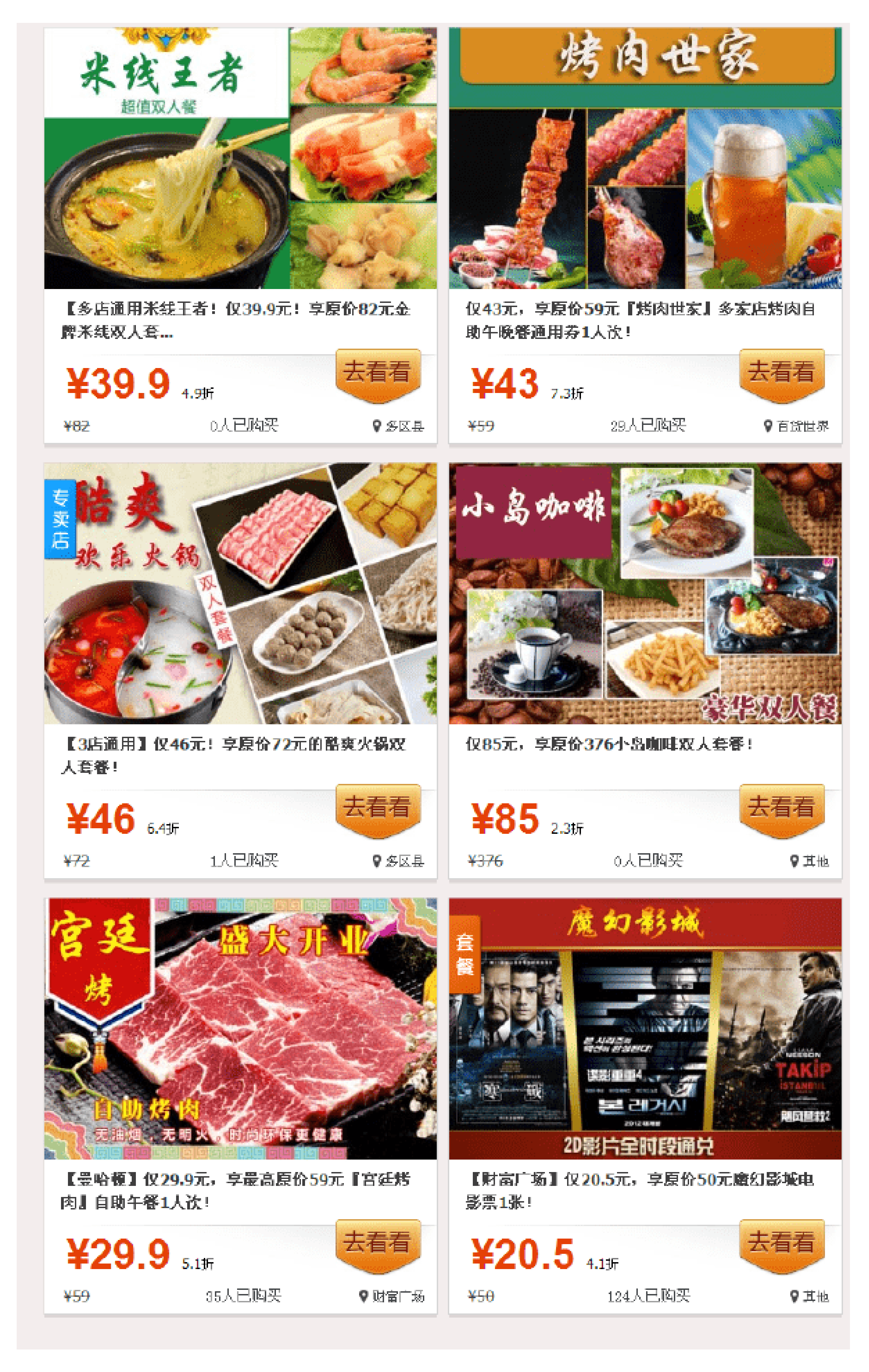


图 24-8　新品展示模块

图 24-8 中只列出了 6 项产品的信息，在运营中的团购网站首页新品展示可能多达几十个，甚至上百个，但是每个产品的实现代码都很相似，只要掌握了本实例中的代码，完全可以完成显示需求。

实现本节模块的具体代码如下。

```
<ul class=goods_listind>
  <li class=goods_listindli>
  <h2><a class="spti_a yahei" title=米线王者 href="#" link=_blank data-prarm="click_channel1_all_title-0-d7fb83afb45efafb">【多店通用米线王者！仅 39.9 元！享原价 82 元金牌米线双人套...</a>
  </h2><a class=picture href="#" link=_blank data-prarm="click_channel1_all_img-0-d7fb83afb45efafb">
  <img class=sp_img src="images/goods_1345621705_2674_1.jpg" width=358 height=238> </a>
  <div style="padding-bottom: 0px; padding-left: 15px; width: 268px; padding-right: 75px; padding-top: 0px" class="buy_boxind clearfix">
  <a class="bh buy_a" href="#" link=_blank data-prarm="click_channel1_all_button-0-d7fb83afb45efafb">去看看</a> <span class=num>¥39.9</span> <em>4.9 折</em> </div>
  <ul>
    <li class="left yahei">¥82</li>
    <li class=center data-id="d7fb83afb45efafb">0 人已购买</li>
    <li class=right><span>多区县</span></li></ul>
  <div class=sp_yy>ico</div></li>
  <li class=goods_listindli>
  <h2><a class="spti_a yahei" title=烤肉世家 href="#" link=_blank data-prarm="click_channel1_all_title-1-0d20024fa823b3a8">仅 43 元，享原价 59 元『烤肉世家』多家店烤肉自助午晚餐通用券 1 人次！</a>
  </h2><a class=picture href="#" link=_blank data-prarm="click_channel1_all_img-1-0d20024fa823b3a8"><img class=sp_img src="images/goods_1349943361_7356_1.jpg" width=358 height=238> </a>
  <div style="padding-bottom: 0px; padding-left: 15px; width: 268px; padding-right: 75px; padding-top: 0px" class="buy_boxind clearfix">
  <a class="bh buy_a"  href="#" link=_blank data-prarm="click_channel1_all_button-1-0d20024fa823b3a8">去看看</a> <span class=num>¥43</span> <em>7.3 折</em> </div>
  <ul>
    <li class="left yahei">¥59</li>
    <li class=center data-id="0d20024fa823b3a8">29 人已购买</li>
    <li class=right><span>百货世界</span></li></ul>
  <div class=sp_yy>ico</div></li>
  <li class=goods_listindli>
  <h2><a class="spti_a yahei" title=酷爽火锅！ href="#" link=_blank  data-prarm="click_channel1_all_title-2-d4f6a300af785a9c">【3 店通用】仅 46 元！享原价 72 元的酷爽火锅双人套餐！</a>
  </h2><a class="li_indlogo bh" href="#" link=_blank>专卖店</a>
      <a class=picture href="#" link=_blank data-prarm="click_channel1_all_img-2-d4f6a300af785a9c"><img class=sp_img src="images/goods_1349851399_5923_1.jpg"  width=358 height=238> </a>
  <div style="padding-bottom: 0px; padding-left: 15px; width: 268px; padding-right: 75px; padding-top: 0px"  class="buy_boxind clearfix">
  <a class="bh buy_a" href="#" link=_blank data-prarm="click_channel1_all_button-2-d4f6a300af785a9c">去看看</a> <span class=num>¥46</span> <em>6.4 折</em> </div>
  <ul>
```

```
            <li class="left yahei">¥72</li>
            <li class=center data-id="d4f6a300af785a9c">1 人已购买</li>
            <li class=right><span>多区县</span></li></ul>
        <div class=sp_yy>ico</div></li>
        <li class=goods_listindli>
        <h2><a class="spti_a yahei" title=小岛咖啡 href="#" link=_blank  data-prarm="click_channel1_all_titl
e-3-561002cba5b24ac6">仅 85 元，享原价 376 小岛咖啡双人套餐！</a>
        </h2><a class=picture  href="#" link=_blank  data-prarm="click_channel1_all_img-3-561002cba5b24ac
6">
       <img class=sp_img src="images/goods_1349926783_2203_1.jpg"  width=358 height=238> </a>
        <div style="padding-bottom: 0px; padding-left: 15px; width: 268px; padding-right: 75px; padding-top:
 0px" class="buy_boxind clearfix">
       <a class="bh buy_a"  href="#" link=_blank data-prarm="click_channel1_all_button-3-561002cba5b24ac6
">去看看</a> <span class=num>¥85</span> <em>2.3 折</em> </div>
        <ul>
            <li class="left yahei">¥376</li>
            <li class=center data-id="561002cba5b24ac6">0 人已购买</li>
            <li class=right><span>其他</span></li></ul>
        <div class=sp_yy>ico</div></li>
        <li class=goods_listindli>
        <h2><a class="spti_a yahei" title=自助烤肉！ href="#" link=_blank data-prarm="click_channel1_all_titl
e-4-6f344b02f069ebb8">【曼哈顿】仅 29.9 元，享最高原价 59 元『宫廷烤肉』自助午餐 1 人次！</a>
        </h2><a class=picture href="#" link=_blank data-prarm="click_channel1_all_img-4-6f344b02f069ebb8">
<img class="sp_img lazyload" src="images/grey.gif" width=358 height=238 original="images/goods_134985222
0_2880_1.jpg">
        </a>
        <div style="padding-bottom: 0px; padding-left: 15px; width: 268px; padding-right: 75px; padding-top:
 0px" class="buy_boxind clearfix"><a class="bh buy_a" href="#" link=_blank  data-prarm="click_channel1_all
_button-4-6f344b02f069ebb8">去看看</a>
       <span class=num>¥29.9</span> <em>5.1 折</em> </div>
        <ul>
            <li class="left yahei">¥59</li>
            <li class=center data-id="6f344b02f069ebb8">35 人已购买</li>
            <li class=right><span>财富广场</span></li></ul>
        <div class=sp_yy>ico</div></li>
        <li class=goods_listindli>
        <h2><a class="spti_a yahei" title=魔幻影城！ href="#" link=_blank data-prarm="click_channel1_all_titl
e-5-46b628c4e80e6ea4">【财富广场】仅 20.5 元，享原价 50 元魔幻影城电影票 1 张！</a>
        </h2><a class="li_indlogo bh tc_index" href="#" link=_blank>套餐</a>
       <a class=picture href="#" link=_blank   data-prarm="click_channel1_all_img-5-46b628c4e80e6ea4">
      <img class="sp_img lazyload" src="images/grey.gif" width=358 height=238   original="images/goods_1
349853854_5414_1.jpg">
```

```
        </a>
        <div style="padding-bottom: 0px; padding-left: 15px; width: 268px; pa
dding-right: 75px; padding-top: 0px" class="buy_boxind clearfix">
            <a class="bh buy_a"  href="#" link=_blank  data-prarm="click_cha
nnel1_all_button-5-46b628c4e80e6ea4">去看看</a><span class=num>¥20.5</spa
n> <em>4.1 折</em>
       </div>
        <ul>
            <li class="left yahei">¥50</li>
            <li class=center data-id="46b628c4e80e6ea4">124 人已购买</li>
            <li class=right><span>其他</span></li></ul>
        <div class=sp_yy>ico</div></li>
    </ul>
```

以上代码共展示了 6 种商品的信息。

24.2.5 侧边栏模块分析

网页主体右侧为侧边栏，主要包括热门分类模块、热销商品排行模块和热门频道模块。具体效果如图 24-9 所示。

以上模块的实现代码如下所示。

1. 侧边栏框架代码

```
<div class=con_boxrig>
```

2. 热门分类代码

```
<div id=all_seerig>
<h2 class=yahei>热门分类</h2>
<ul class=clearfix>
    <li><a href="#"
    data-prarm="click_channel1r_dianying">电影</a> </li>
    <li><a href="#"
    data-prarm="click_channel1r_17-0-0-0-0-1">自助餐</a> </li>
    <li><a href="#"
    data-prarm="click_channel1r_40-0-0-0-0-1">足疗按摩</a> </li>
    <li><a href="#"
    data-prarm="click_channel1r_57-0-0-0-0-1">食品保健</a> </li>
    <li><a href="#"
    data-prarm="click_channel1r_33-0-0-0-0-1">美发</a> </li>
    <li><a href="#"
    data-prarm="click_channel1r_44-0-0-0-0-1">汽车服务</a> </li>
    <li><a href="#"
    data-prarm="click_channel1r_25-0-0-0-0-1">运动健身</a> </li>
```



图 24-9 侧边栏模块

```
        <li><a href="#"
        data-prarm="click_channel1r_27-0-0-0-0-1">游乐游艺</a> </li>
        <li><a href="#"
        data-prarm="click_channel1r_35-0-0-0-0-1">美容塑形</a> </li></ul></div>
```

以上代码主要使用了<ul>标签构成文字序列，然后使用<a>标签为每一个分类做超链接。

3. 热销商品排行榜代码

```
    <div id=rightrank>
    <h2 class=yahei>热销商品排行榜</h2>
    <ul>
        <li class=on>
        <div class=tjshow><b class=one>ico</b> <a href="#" link=_blank data-prarm="click_channelr1-hot-img
-0-a3b5c0fb643f099d"><img class=pd_img alt=qq！ src="images/18goods_1334053028_9657_3.jpg">
    </a>
        <div class=ritbox><em class="one yahei">¥</em><em
        class="two yahei">99</em><br><em class=three>1893</em>人购买 </div>
        <p><a href="#" link=_blank data-prarm="click_channelr1-hot-title-it_index-a3b5c0fb643f099d">lakrina
春秋被</a></p></div></li>
        <li class=on>
        <div class=tjshow><b class=two>ico</b> <a href="#" link=_blank data-prarm="click_channelr1-hot-im
g-1-48456be593e0dcba"><img class=pd_img alt=qq src="images/11goods_1334201980_7392_3.jpg">
    </a>
        <div class=ritbox><em class="one yahei">¥</em><em class="two yahei">49</em><br><em class=thre
e>21193</em>人购买 </div>
        <p><a href="#" link=_blank data-prarm="click_channelr1-hot-title-it_index-48456be593e0dcba">时尚男
士拉链钱包</a></p></div></li>
        <li class=on data="recommend"><span class=three>瘦身纤体梅</span>
        <div class=tjshow><b class=three>ico</b> <a href="#" link=_blank data-prarm="click_channelr1-hot-i
mg-2-c0f41f265f1ebf80"><img class=pd_img alt=qq src="images/13goods_1334123797_4501_3.jpg">
    </a>
        <div class=ritbox><em class="one yahei">¥</em><em class="two yahei">1</em><em
        class="one yahei">.99</em><br><em class=three>270323</em>人购买 </div>
        <p><a href="#" link=_blank data-prarm="click_channelr1-hot-title-it_index-c0f41f265f1ebf80">瘦身纤体
梅</a></p></div></li>
        <li data="recommend"><span class=four>抛弃型过滤烟嘴</span>
        <div class=tjshow><b class=four>ico</b> <a href="#" link=_blank data-prarm="click_channelr1-hot-im
g-3-eb81051c35f789bf"><img class=pd_img alt=qq！ src="images/12goods_1333255335_3909_3.jpg">
    </a>
        <div class=ritbox><em class="one yahei">¥</em><em class="two yahei">1</em><em class="one yahe
i">.9</em><br><em class=three>39922</em>人购买 </div>
        <p><a href="#" link=_blank data-prarm="click_channelr1-hot-title-it_index-eb81051c35f789bf">抛弃型
```

```
过滤烟嘴</a></p></div></li>
        <li data="recommend"><span class=five>男士单肩斜挎包</span>
        <div class=tjshow><b class=five>ico</b>
          <a href="#" link=_blank data-prarm="click_channelr1-hot-img-4-48424ef428f34dd5"><img class=pd
_img alt=qq src="images/13goods_1333259469_7784_3.jpg">
          </a>
        <div class=ritbox><em class="one yahei">¥</em><em class="two yahei">78</em><br><em class=thre
e>1256</em>人购买 </div>
        <p><a href="images" link=_blank data-prarm="click_channelr1-hot-title-it_index-48424ef428f34dd5">男
士单肩斜挎包</a></p></div></li>
        <li data="recommend"><span class=six>家纺保健枕</span>
        <div class=tjshow><b class=six>ico</b>
           <a href="#" link=_blank data-prarm="click_channelr1-hot-img-5-da9df721d6ff6d82">
           <img class=pd_img alt=qq！ src="images/13goods_1333258035_7250_3.jpg">
           </a>
        <div class=ritbox><em class="one yahei">¥</em><em class="two yahei">36</em><br><em class=thre
e>1864</em>人购买 </div>
        <p><a href="#" link=_blank data-prarm="click_channelr1-hot-title-it_index-da9df721d6ff6d82">家纺保
健枕</a></p></div></li>
        <li data="recommend"><span class=seven>超柔亲肤空调夏被</span>
        <div class=tjshow><b class=seven>ico</b>
           <a href="#" link=_blank data-prarm="click_channelr1-hot-img-6-b50c9aebc4ae727a">
           <img class=pd_img alt=qq！ src="images/11goods_1333163649_1180_3.jpg">
           </a>
        <div class=ritbox><em class="one yahei">¥</em><em class="two yahei">59</em><br><em class=thre
e>640</em>人购买 </div>
        <p><a href="#" link=_blank data-prarm="click_channelr1-hot-title-it_index-b50c9aebc4ae727a">超柔亲
肤空调夏被</a></p></div></li>
        <li data="recommend"><span class=eight>环保印花活性四件套</span>
        <div class=tjshow><b class=eight>ico</b>
           <a href="#" link=_blank data-prarm="click_channelr1-hot-img-7-91be5abbb374b770">
           <img class=pd_img alt=qq！ src="images/18goods_1333017091_3843_3.jpg">
           </a>
        <div class=ritbox><em class="one yahei">¥</em><em class="two yahei">95</em><br><em class=thre
e>6577</em>人购买 </div>
        <p><a href="#" link=_blank data-prarm="click_channelr1-hot-title-it_index-91be5abbb374b770">环保印
花活性四件套</a></p></div></li>
        <li data="recommend"><span class=nine>加厚真空压缩袋套装</span>
        <div class=tjshow><b class=nine>ico</b>
           <a href="#" link=_blank data-prarm="click_channelr1-hot-img-8-2aa082206312b1a7">
           <img class=pd_img alt=qq src="images/goods_1335508647_4181_3.jpg">
           </a>
```

```
<div class=ritbox><em class="one yahei">¥</em><em class="two yahei">69</em><br><em class=three>5042</em>人购买 </div>
<p><a href="#" link=_blank data-prarm="click_channelr1-hot-title-it_index-2aa082206312b1a7">加厚真空压缩袋套装</a></p></div></li>
<li data="recommend"><span class=ten>美佳 2 件套</span>
<div class=tjshow><b class=ten>ico</b>
  <a href="#" link=_blank data-prarm="click_channelr1-hot-img-9-e685d369ba834f4a">
  <img class=pd_img alt=qq src="images/16goods_1333009457_2889_3.jpg">
  </a>
<div class=ritbox><em class="one yahei">¥</em><em class="two yahei">48</em><br><em class=three>10590</em>人购买 </div>
<p><a href="#" link=_blank data-prarm="click_channelr1-hot-title-it_index-e685d369ba834f4a">美佳 2 件套</a></p></div></li></ul></div>
```

4. 热门频道代码

```
<div id=channelrirht class=yahei>
<h2 class=hd>热门频道</h2>
<ul>
  <li><a class=meishi href="#" data-prarm="click_channelr1"><span><strong>美食</strong> 中餐/火锅/自助餐<br>特色餐饮/蛋糕... </span><em>美食</em> </a></li>
  <li><a class=yule href="#" data-prarm="click_channelr2"><span><strong>娱乐</strong>ktv/游乐游艺/温泉<br>运动健身/演出... </span><em>娱乐</em> </a></li>
  <li><a class=dianying href="#" data-prarm="click_channelr3"><span><strong>电影</strong> 低价看大片， 精彩<br>别错过 </span><em>电影</em> </a></li>
  <li><a class=meirong href="#" data-prarm="click_channelr4"><span><strong>美容保健</strong> 美发/足疗按摩/美甲<br>美容塑性/养生... </span><em>美容保健</em> </a></li>
  <li><a class=life href="#" data-prarm="click_channelr5"><span><strong>生活服务</strong> 摄影写生/母婴亲子<br>汽车服务/教育... </span><em>生活服务</em> </a></li>
  <li><a class=shop href="#" data-prarm="click_channelr8"><span><strong>网购</strong> 服装/日用家居/食品<br>保健/个护化妆... </span><em>网购</em> </a></li></ul></div>
```

以上代码实现了热门频道模块的功能。

```
<div style="width: 222px" id=floatbox>
<div id=floatad></div></div>
</div>
```

24.2.6 热门城市索引模块分析

每个城市都有自己的团购内容，为了方便浏览者跨区域访问团购信息，在主体下方设置了一个热门城市索引模块，该模块主要以文字和超链接实现，具体效果如图 24-10 所示。

热门城市：北京团购 深圳团购 无锡团购 天津团购 沈阳团购 济南团购 郑州团购 石家庄团购 成都团购 上海团购 南京团购 长沙团购
西安团购 广州团购 杭州团购 青岛团购 大连团购 宁波团购 苏州团购 重庆团购 武汉团购 厦门团购 哈尔滨团购 合肥团购

图 24-10 热门城市索引模块

该模块结构简单，具体实现代码如下。

```
<div class=hot_city>
<dl class=city_dl>
  <dd class="city_dd clearfix"><strong class=hot_citystr>热门城市：</strong>
 <a class=hot_citya href="#">北京团购</a>
 <a class=hot_citya href="#">深圳团购</a>
 <a class=hot_citya href="#">无锡团购</a>
 <a class=hot_citya href="#">天津团购</a>
 <a class=hot_citya href="#">沈阳团购</a>
 <a class=hot_citya href="#">济南团购</a>
 <a class=hot_citya href="#">郑州团购</a>
 <a class=hot_citya href="#">石家庄团购</a>
 <a class=hot_citya href="#">成都团购</a>
 <a class=hot_citya href="#">上海团购</a>
 <a class=hot_citya href="#">南京团购</a>
 <a class=hot_citya href="#">长沙团购</a>
 <a class=hot_citya href="#">西安团购</a>
 <a class=hot_citya href="#">广州团购</a>
 <a class=hot_citya href="#">杭州团购</a>
 <a class=hot_citya href="#">青岛团购</a>
 <a class=hot_citya href="#">大连团购</a>
 <a class=hot_citya href="#">宁波团购</a>
 <a class=hot_citya href="#">苏州团购</a>
 <a class=hot_citya href="#">重庆团购</a>
 <a class=hot_citya href="#">武汉团购</a>
 <a class=hot_citya href="#">厦门团购</a>
 <a class=hot_citya href="#">哈尔滨团购</a>
 <a class=hot_citya href="#">合肥团购</a>
 </dd></dl></div>
```

24.2.7 底部模块分析

底部主要是客户服务和快捷链接模块，用于解决各种客户服务问题，具体效果如图 24-11 所示。

用户帮助
玩转阿里
常见问题
秒杀规则
积分规则
消费者保障
网站地图

获取更新
阿里团新浪微博
阿里团开心网主页
阿里团豆瓣小组
RSS订阅
手机版下载

商务合作
商家入驻
提供团购信息
友情链接
开放API

公司信息
关于我们
媒体报道
加入我们
隐私声明
用户协议

24小时服务热线
500-000-0000
500-000-0000
我要提问

图 24-11　底部模块

该模块内容主要也是文字和超链接，实现起来较简单，具体代码如下。

```
<div id=footer><b class=top>ico</b>
<div class="bottom_box clearfix">
<ul class=boul_list>
  <li class=li_x>
  <h2 class=yahei>用户帮助</h2></li>
  <li><a class=bolist_a href="#">玩转阿里</a></li>
  <li><a class=bolist_a href="#">常见问题</a></li>
  <li><a class=bolist_a href="#">秒杀规则</a></li>
  <li><a class=bolist_a href="#">积分规则</a></li>
  <li><a class=bolist_a href="#">消费者保障</a></li>
  <li><a class=bolist_a href="#">网站地图</a></li></ul>
<ul class=boul_list>
  <li class=li_x>
  <h2 class="h2_1 yahei">获取更新</h2></li>
  <li><a class=bolist_a href="#" link=_blank data-prarm="weibo">阿里团新浪微博</a></li>
  <li><a class=bolist_a href="#" link=_blank data-prarm="kaixin">阿里团开心网主页</a></li>
  <li><a class=bolist_a href="#" link=_blank data-prarm="douban">阿里团豆瓣小组</a></li>
  <li><a class=bolist_a href="#" data-prarm="rss">rss 订阅 </a></li>
  <li><a class=bolist_a href="#" data-prarm="click_mobile_bottom">手机版下载</a></li></ul>
<ul class=boul_list>
  <li class=li_x>
  <h2 class="h2_2 yahei">商务合作</h2></li>
  <li><a class=bolist_a href="#">商家入驻</a></li>
  <li><a class=bolist_a href="#">提供团购信息</a></li>
  <li><a class=bolist_a href="#">友情链接</a></li>
  <li><a class=bolist_a href="#">开放 api </a></li></ul>
<ul class=boul_list>
  <li class=li_x>
  <h2 class="h2_3 yahei">公司信息</h2></li>
  <li><a class=bolist_a href="#">关于我们</a></li>
  <li><a class=bolist_a href="#">媒体报道</a></li>
  <li><a class=bolist_a href="#">加入我们</a></li>
```

```
        <li><a class=bolist_a href="#">隐私声明</a></li>
        <li><a class=bolist_a href="#">用户协议</a></li></ul>
    <div class=kefu_bottom><!--<h2 class="yahei"><a href="#"title="阿里团在线客服">阿里团在线客服</a>
</h2>
                    <span class="bh wan_x">横线</span>-->
    <h2 class="kh2_1 yahei">24 小时服务热线</h2>
    <h2 class="kh2_2 yahei">500-000-0000</h2>
    <h2 class="kh2_2 yahei">500-000-0000</h2><!--<span class="bh wan_x">横线</span>-->
    <a class="bh kfwwweibo" href="#" link=_blank>阿里团客服微博</a>
    <h2 class="kh2_3 yahei">
    <a href="#" link=_blank>我要提问</a></h2></div></div></div>
```

24.3　调整

网站设计完成后，如果需要完善或者修改，可以对其中的框架代码以及样式代码进行调整，下面简单介绍几项内容的调整方法。

24.3.1　部分内容调整

要修改网页整体背景色，可打开 css 样式表，修改 body 标记样式，修改后代码如下。

```
body {
    background:#ccffff;
}
```

使用 Photoshop CS6 绘制 3 个图像，logo.gif、dhbg1.gif 和 dhbg2.gif。其中 dhbg2.gif 比 dhbg1.gif 的渐变色深一些，如图 24-12 所示。



图 24-12　绘制的 3 个图像

修改样式表中的 headNav 样式属性，修改后代码内容如下。

```
#headnav {
 background:url(../images/dhbg2.gif) no-repeat
}
#headnav ul li a {
 background:url(../images/dhbg1.gif) no-repeat
}
#headnav ul li a b {
 background:url(../images/dhbg1.gif) no-repeat
}
```

24.3.2 模块调整

在本实例中，网站模块基本固定，模块位置基本不需要调整，如果要修改的话，可以在网页主体内容前增加一个 banner，在导航栏下方插入 banner 代码。

```
<div style="height:10px"></div> //插入一个高度为 10px 的空模块，进行上下内容分隔
<div style="text-align:center">
<a href="#"><img src="images/banner.jpg"></a>
</div>
```

24.3.3 调整后预览测试

调整后，网页预览效果如图 24-13 所示。



图 24-13 网页的整体效果